A REVISTA DOS USUÁRIOS DO TK

MICROHOBBY

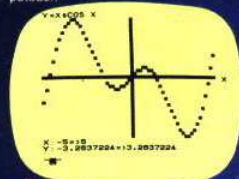
Ano I – Nº 4 – 1983

Por Dentro do Apple

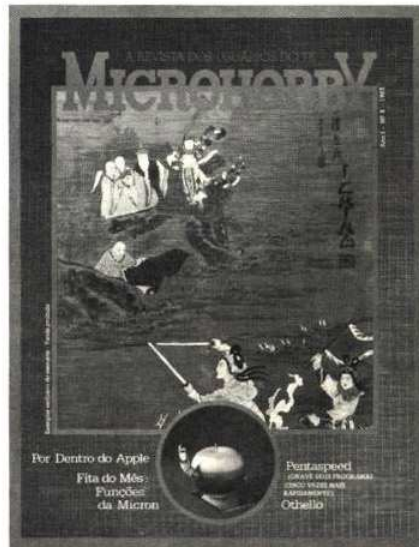
Fita do Mês : Funções da Micron

Pentaspeed (GRAVE SEUS PROGRAMAS CINCO VEZES MAIS RAPIDAMENTE)

Othello

MICROSOFT
Programas para o seu TK82-C e TK85
JOGO DE GAMÃO
16K
Este programa apresenta o tabuleiro no vídeo e utiliza o eficiente código de máquina, permitindo 4 (quatro) níveis de dificuldades de jogo.
MONSTRO DAS TREVAS TRIDIMENSIONAL · 16K
Impressionante jogo onde você deve evitar o monstro das trevas. Tudo em 3 dimensões.
DEMOLIDOR
2K
Jogo animado, tipo "fliperama". O jogador deverá demolir uma parede, com uma bola que se encontra sempre em movimento.
LABIRINTO TRIDIMENSIONAL · 16K
Jogo em três dimensões. O jogador poderá definir a dificuldade do Labirinto. O programa apresenta a posição do jogador em perspectiva. Em qualquer momento é possível pedir auxílio ao computador.
INVASORES DO ESPAÇO · 16K
Consiste de uma frota de naves invasoras extraterrenas, descendo no planeta Terra. Sua missão é destruir as naves invasoras dispondo da arma de raios laser.
RALLY
16K
Emocionante corrida de rally em um labirinto, onde poderão ser testados sua habilidade e seus reflexos. Para conseguir seu intento, você deverá evitar carros-ataque e obstáculos em seu trajeto.
MATEMÁTICA I
16K/64K
Análise gráfica de funções matemáticas, resolução de sistemas de equações lineares (16K-51 equações/64K-95 equações), e Cálculo de integrais definidos.
TK-MAN
16K
Jogo animado onde deverão ser apagados todos os pontinhos espalhados em um labirinto (o programa contém 15 tipos de labirinto). Você será impedido a qualquer custo, por 4 extraterrenos, guardiães do labirinto, que poderão ser combatidos com cargas de raios-laser.
T-KALC
16K/64K
Programa desenvolvido para cálculos numéricos em planilha. O usuário define as colunas, as linhas e as fórmulas aplicadas. Similar ao famoso Visicalc. De grande versatilidade, este programa permite a formulação de cálculos científicos e comerciais, análise de tabelas numéricas e outras aplicações.
TKADREZ II
16K
Este jogo apresenta o tabuleiro e as peças no vídeo. Permite a escolha de até 7 níveis de dificuldade. O programa fornece a qualquer momento, a listagem dos lances efetuados, e armazena em fita a posição das peças. Ele poderá recomendar a sua jogada.
NOS REVENDEDORES AUTORIZADOS EM TODO PAÍS
80 programas à sua disposição. Solicite folheto.
MICROSOFT®
Av. Angélica, 2.318 - 13º - Cj. 132
Tel.: (011) 256-3858 - CEP 01228 - São Paulo - SP

Expediente

DIRETOR E EDITOR:
Pierluigi Piazzi
REDAÇÃO:
Jornalista Responsável:
Aristides Ribas de Andrade Fº
Coordenação Editorial:
Ana Lúcia de Alcântara
Analistas de Software:
Nancy Mitie Ariga, Roberto Bertini Renzetti, Carlos Eduardo Rocha Salvato
COLABORARAM NESTE NÚMERO:
Wilson José Tucci, Renato da Silva Oliveira, Bernardo C. Stein, Edson Mikio Yoshida, Nelson Murassaki, Fátima M.R. Gouveia, Eliana Santos Queiroz, Osmère Sarkis, Rosa K. Fromer, Betty Feffer.
ASSESSORIA TÉCNICA:
Flavio Rossini, Cassiano Roda
CORRESPONDENTES:
Londres — Robert L. Lloyd
Paris — Alain Richard
N. York — Natan Portnoy
Milão — Bruno Origo
DEPARTAMENTO ADMINISTRATIVO:
Assistente: Rosana S. Mariano
DEPTO. COMERCIAL:
Gerente Comercial:
Gina Elimelek
Assistente: Atilio Debatin
Publicidade: Aurio J. Mosolino (sup.)
Lídia Pauluk, Edson R. Silva (contatos)
Rosângela A. Gomes (secr.) - tel.: 256-8348
Assinaturas: Carlos H. Oliveira
tel.: 257-5767
FOTOLITO:
Flash Color
IMPRESSÃO:
Gráfica Castelo
TIRAGEM:
30.000 exemplares
ASSINATURAS:
12 meses: Cr$ 11.800,00
MICROHOBBY
é uma publicação mensal da Micromega Public. Mat. Didático Ltda.
INPI 2992 - Livro A
CORRESPONDÊNCIA:
Caixa Postal 60081 — CEP 05096 — SP
ESCRITÓRIO CENTRAL:
R. Bahia, 1049 — CEP 01244 — São Paulo
Tels.: 257-5767 e 256-8348
Distribuição só para assinantes.

Índice

CAPA: Hugo Faleiros e Cassiano Roda

Editorial

Nem só de pão vive o homem: de elogios também. Através de nossos correspondentes e viagens de amigos, vários exemplares de MICROHOBBY já estão circulando pelos Estados Unidos e pela Europa. Isto sem contar alguns assinantes no exterior: poucos, mas motivo de muito orgulho!

Pois bem, é incrível a reação quase unânime de espanto do pessoal lá de fora ao ver a qualidade gráfica e editorial da **nossa** *revista (quando dizemos* **nossa**, *você está incluído!).*

Isso nos envaidece muito e queremos compartilhar com você este sentimento de orgulho.

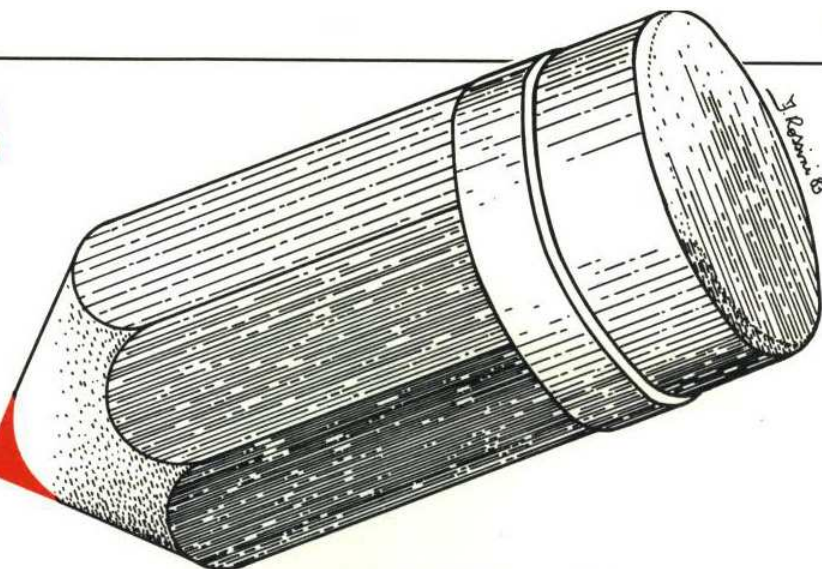
Para o aprimoramento desta qualidade contribuiram de maneira significativa as inúmeras cartas contendo preciosas sugestões e críticas, e as colaborações de altíssimo nível de muitos leitores.

Através destas cartas, por exemplo, sentimos a necessidade de abrir mais uma seção: POR DENTRO DO APPLE. Muitíssimos leitores têm um TK em casa, mas na firma têm acesso a um UNITRON, um MICROENGENHO, um MAXXI, etc., todos compatíveis com o APPLE e gostariam de receber algumas dicas referentes a estes outros computadores.

Outros leitores, pensando talvez numa profissionalização do HOBBY aprendido com o TK, nos solicitaram informações a respeito de computadores de maior porte. Isto nos motivou, inclusive, a pensar em abrir uma seção irmã à do APPLE dedicada aos computadores compatíveis com o TRS80.

Tudo isso, obviamente, representará algo mais para o usuário do TK a quem dedicamos nossa revista, não implicando de forma alguma em perda de qualidade do material que ele recebe para seu pequeno grande computador.

Por isso, pedimos: continuem escrevendo criticando, cobrando, colaborando, sugerindo, ou até apenas elogiando: afinal nem só de pão vive o homem. . .

CARTA DOS LEITORES

Prezados Senhores,

Em primeiro lugar quero parabenizar a MICROMEGA pela excelente publicação mensal da MICROHOBBY, pois é uma revista que atende aos anseios de todos os usuários do TK, destacando-se os programas inéditos, as dicas e as novidades.

Tenho algumas sugestões que acredito melhorariam ainda mais o nível da revista, que são: poderia ser criada uma seção para solucionar as dúvidas dos usuários do TK, tanto em software como em hardware; gostaria que fosse publicado um programa que informasse ao usuário o quanto de memória ainda resta durante a digitação de um programa extenso, este programa poderia ser uma sub-rotina do programa principal, ao qual o usuário recorreria sempre que achasse necessário.

Desde já agradeço a atenção dispensada,

Silvio Roberto Hantschel

Prezado Silvio,

Uma de suas sugestões já foi atendida: aconselhamos a leitura do MAPA DA RAM do TK na seção de Dicas e da análise da FITA DO MÊS.

Prezados Senhores,

Estou enviando anexo a esta, um programa com os comentários, para participar da seção "Quebra-cabeça".

Aproveito a oportunidade para solicitar alguns esclarecimentos. . . Os programas que recebi na fita-brinde necessitam que o computador funcione em "SLOW". Meu computador (NEZ8000) só funciona em FAST. Na revista nº Ø, pag. 6 foi anunciado que na revista nº 1 seria publicado artigo mostrando como acrescentar a função "SLOW" ao computador. Porque não foi publicado? Vocês não podem enviar instruções para eu fazer a modificação? Agradeço resposta, despeço-me. Atenciosamente.

Hermelindo Pinheiro Manoel

Prezado Hermelindo,

Seu programa e comentários estão ótimos. Infelizmente ainda não foi para esta vez sua publicação. Continue, enviando respostas pois o nível desta última estava muito boa.

Com relação ao SLOW no NEZ8000 sugerimos uma lida na seção PEQUENOS ANÚNCIOS: o WILSON DE ASSIS pode lhe fazer esta adaptação ou o JAN MARTIN LUND pode lhe fornecer o esquema.

Prezados Senhores,

Assinante que sou de MICROHOBBY, no final de julho (28) recebi satisfeito, a fita brinde prometida na assinatura.

Achei interessante o jogo da "Pulga", entretanto, por mais que tentasse em todos os volumes, não consegui rodar o "Simulador". Seguramente não está boa a fita.

Dois ou três dias após foi-me entregue pelo Correio, o nº 2 da revista (excelente) e nele verifiquei a opção permitida aos novos assinantes; o desgravador Twin Go. Li também a informação de que o cassete estaria gravado dos dois lados, o que também não ocorre com o meu como poderão constatar.

Em razão do acima exposto remeto-vos anexa a esta, a fita em questão, para que seja examinada.

Julgando-me no direito de receber um cassete com os dois lados gravados bons, como prometido na proposta de assinante, permito-me a liberdade de solicitar de V.Sas., desde que possível e sem prejuizo para a revista, a troca de brinde. Apesar da curiosidade muito grande em torno do Simulador que não conheço, para as minhas aplicações, me seria muito mais útil o Twin Go.

Outrossim, aproveito o ensejo para perguntar como posso medir a capacidade de minha expansão de 64K, uma vez que, aplicando : 999Ø PRINT PEEK 16388 + + 256 * PEEK 16389 – 16384 encontro: 48.896 Bytes. Certo da compreensão de V.Sas., atenciosamente.

Lorenzo C.B. Scaffa Falcão

Prezado Lorenzo,

Infelizmente nosso estoque de TWIN GO já está inteiramente comprometido e não podemos efetuar a troca solicitada. Podemos, porém, trocar a fita do "Simulador de VÔO" por outra idêntica (que você nos devolveu foi testada e não estava com defeito) ou pela nova fita SÃO PAULO e MANSÃO MALUCA. Escreva-nos comunicando sua decisão.

Com relação à sua expansão para 64K aconselhamos a leitura da seção DESGRILANDO.

A EXPANSÃO PARA 64 K

Prezados Senhores,

Seria injustiça de minha parte, não acusar o recebimento da FITA BRINDE e muito mais, deixar de agradecer-lhes pela mesma. O Programa "Simulador de VÔO" é realmente fantástico. Por isso, e pelo que promete a revista, acho que foi um bom negócio tornar-me assinante da mesma.

Mas nós, amadores do Basic-TK precisamos mesmo de uma publicação de apoio, como promete ser a MICROHOBBY. E no momento, eu, em particular estou pedindo ajuda a vocês. Trata-se de saber se existe algum programa capaz de calcular com precisão a quantidade de memória (em bytes ou Kbytes) ocupada com determinado programa, pois é muito desagradável, ao se digitar um programa longo, ver "estourar" a memória do computador. Assim, podendo "medir" a quantidade de memória ocupada, torna-se prático e mais fácil trabalhar, mesmo quando se tem uma expansão de memória de 64K, como é meu caso. Mas já me aconteceu de perder programas quase no final do serviço. E aí, como se faz? Não adianta "esquentar" a "cuca".

No meu caso, as coisas são piores. Explico: no livro de Délio Santos Lima, "45 PROGRAMAS PRONTOS PARA RODAR EM TK-82C e NE Z8000" (Livraria Poliedro — 3ª edição), há na página 4, umas dicas para se obter o número de bytes ocupados pelo programa, que é a seguinte:

PRINT PEEK 16396 + 256 * PEEK 16397 — 16509

e há também variáveis e display:

PRINT PEEK 16404 + 256 * PEEK 16405 — 16509

Recomendando usar o comando diretamente como indicado, e NEW LINE. No final da listagem do programa, é claro, e sem etiqueta.

Após ter digitado um longo programa, fiz o comando conforme indicado e o resultado foi de 15,960 Kbytes. Com esse resultado, fiquei tranquilo e prossegui com mais umas 8 linhas de programa (+ ou —) e foi aí que aconteceu a BOMBA. O computador limpou tudo e ficou a tela com traços horizontais que depois rodavam na tela, etc, etc. Será que os amigos podem me dizer alguma coisa, se a EXPANSÃO QUE POSSUO É DE 64K? E PORQUE QUANDO O PROBLEMA ESBARROU OU TENTOU ULTRAPASSAR OS 16K, a memória estourou? Ou será que o programa do DÉLIO SANTOS LIMA não está correto? Ou será que a MICRODIGITAL está trapaceando, vendendo gato por lebre? Gostaria que os amigos me ajudassem, pois a memória que vai estourar agora, será com certeza a minha.

Para uma resposta mais rápida, mando um envelope selado. Mas se considerarem esse caso de importância (pode estar acontecendo com outros TK) podem dar a dica através da revista, o que seria de grande utilidade talvez, para outros usuários do TK.

Em próxima oportunidade, voltarei com programas de utilidade COMERCIAL, de minha autoria para colaborar com a Revista MICROHOBBY; no momento, os mesmos estão sendo testados, afim de evitar que surjam problemas ou erros durante a execução.

continua

Por falar em teste, peço a atenção dos amigos, em especial do Sr. PIERLUIGI PIAZZI, para o fato de que estamos aqui desenvolvendo um programa, apelidamos provisoriamente de "GH" (Geração Humana). Parece pura maluquice, é o que todos pensam (quando peço ajuda) quando dizemos que o programa destina-se a fazer a PREVISÃO DO SEXO DO BEBÊ que irá nascer de determinada mãe. A ajuda que pedimos, é que as pessoas forneçam as datas completas de nascimento (só da MÃE, é claro), bem como as datas dos filhos que já houverem nascido. Agora, enquanto estamos em fase de experiência com o programa, queremos saber o percentual de erros e acertos, em 1000 casos, com um desacerto apenas. Para nós, isso já representa alguma coisa. Mas precisamos de mais dados, como conseguir? Preferimos lidar com pessoas desconhecidas. A pesquisa funciona da seguinte forma: A pessoa (Mãe) nos fornece o dia, mês e ano de nascimento, e as datas também dos filhos que já nasceram e o sexo. O computador irá então fazer o cálculo, que irá confirmar o sexo dos que já nasceram, dizendo então qual será o sexo do próximo que irá nascer. O que realmente nos interessa no momento é essa confirmação do sexo dos já nascidos. Pois será esse índice que determinará a validade ou não, do programa.

Creio que os amigos poderiam dar-nos uma "colher de chá", publicando um anúncio pequeno na MICRO-HOBBY pedindo aos usuários do TK, que enviem nome, data de nascimento, do usuário, da esposa e dos filhos todos — inclusive se houver algum já falecido. Ou caso de aborto; as datas desses também devem contar. Esse material poderia ser enviado para:

CRM — Cx. Postal: 43
68250 — Óbidos — PA

Acho que assim, dentro de pouco tempo teríamos mais uma vitória do BASIC TK, além de testar a receptividade da nossa nova revista, que terá que crescer mais, muito mais ainda, e do que temos certeza, essa é a intenção dos Senhores.

Roberto R. Malcher (Óbidos — PA)

Caro Roberto,

Vamos começar pelo fim: você não está ocupando nosso "precioso" tempo. Pelo contrário! Você está dando uma valiosíssima contribuição para a revista e para seus leitores, apresentando os grilos que voce apresentou.

Vamos agora ao seu esclarecimento:

1) MEDIDA DOS BYTES OCUPADOS PELO PROGRAMA — As dicas fornecidas no livro do DÉLIO estão mais que corretas. Para maiores detalhes sobre o significado do que você digitou aconselhamos uma rápida leitura na seção DICAS da matéria **"O MAPA DA RAM DO TK"** *que foi elaborada por causa da sua carta!*

Estude com particular cuidado a variável **RAMTOP** *na dica citada para poder entender o esclarecimento do grilo seguinte.*

A EXPANSÃO PARA 64K — Quando você conecta a expansão para 64K no seu TK82, a memória passa a se organizar como segue:

ENDEREÇO — $1K = 2^{1Ø} = 1Ø24$

ENDEREÇO		
Ø	ROM	ØK
8192	RAM 1	8K
16384	RAM 2	16K / 24K
32768	RAM 3	32K / 4ØK / 48K / 56K
65536		64K

O sistema operacional do TK foi projetado, originalmente, para não aproveitar como RAM os 8 Kbytes marcados como RAM 1. Além disso, as primeiras máquinas construídas tinham uma ROM que previa uma expansão até 16K no máximo, ou seja, a RAM 2 que vai até o endereço 32768. Isto produz três problemas facilmente solucionáveis:

A variável RAMTOP do seu TK assume, provavelmente, o valor máximo de 32768. Quando você conecta sua expansão de 64K o computador "pensa" que tem apenas 16K de RAM e por isso acontecem coisas estranhas ao atingir o limite de 16K.

A solução deste problema é extremamente simples: basta contar ao computador, assim que a memória foi conectada, que ele pode chegar até o endereço 65535. Isto é feito através de um comando direto (conforme indicado no manual de instruções de expansão para 64K).

POKE 16389,255

Ao receber este comando, a variável RAMTOP, cujo valor está armazenado nos bytes de endereços 16388 e 16389, mude seu valor:

	ANTES DO POKE	DEPOIS DO POKE
16388	Ø	Ø
16389	128	255
VALOR DA RAMTOP	32768	6528Ø

Para calcular o valor do **RAMTOP**, *multiplicar o valor armazenada no byte 16389 por 256 e somar o valor contido no byte 16388.*

Se você quiser, portanto, todos os bytes a que tem direito, digite também

POKE 16388,255

que o valor de **RAMTOP** *sobe de 6528Ø para 65535.*

continua

Quando você for elaborar programas específicos para serem usados com expansão de 64K é conveniente colocar os dois POKEs como primeiras instruções do programa, no caso do usuário ter um dos primeiros TKs.

O segundo problema é referente aos 8K da RAM1. O sistema operacional do TK simplesmente os ignora pois ele organiza a RAM a partir do endereço 16384. Isto pode ser considerado, a primeira vista, como uma desvantagem, mas com um pouco de cuca e habilidade pode ser transformado numa coisa maravilhosa: Quando se dá um **NEW** *ou um* **LOAD**, *toda a RAM é apagada, mas o que estiver na RAM 1* **está a salvo**! *Você pode, por exemplo, transferir para este esconderijo um arquivo de um programa, carregar de fita um segundo programa apagando o primeiro e utilizar o arquivo no segundo. Pode armazenar uma dúzia de telas para serem chamadas oportunamente, e assim por diante. Obviamente a utilização mais proveitosa deste nicho de 8K é a localização de rotinas em linguagem de máquina, expandindo o próprio sistema operacional do TK.*

É claro que só podemos escrever nestes 8K através de POKE e só podemos ler através de **PEEK.**

O terceiro problema é referente ao arquivo da imagem: ele não pode ficar "a cavalo" do endereço 32768. Isto significa que ele não pode estar armazenado parte na RAM 2 e parte na RAM 3.

Quem desloca a posição do arquivo da imagem na memória é o comprimento do programa. Quando este é muito extenso, o arquivo pode ficar na posição crítica citada (talvez essa seja a causa do seu problema).

A solução é encompridar artificialmente o programa em algumas centenas de bytes (com uma linha **REM**, *por exemplo) de maneira a fazer o arquivo de imagem cair inteiro na RAM 3.*

3) PROGRAMA COMERCIAL E ESTATÍSTICO – Estamos esperando ansiosamente seu programa COMERCIAL (e os leitores também), enquanto nadarmos num mar de FORCA, SENHA, JOGO DA VELHA e DESENHANDO NA TELA!

Com relação ao programa GH achamos a proposta tão interessante e merecedora de apoio que preparamos um apelo especial na pág. 33 .

Esperamos que estas respostas tenham sido úteis para você e para os leitores com problemas análogos. Um abraço.

PIERLUIGI PIAZZI

PENTASPEED

Muitos felizes possuidores de um TK82 começaram a ser roídos pelo verde micróbio da inveja quando viram um TK85 gravando e lendo programas a uma velocidade incrível, usando o **HIGH-SPEED**. Esta inveja se torna mais intensa quando percebem que não se trata só de alterar a velocidade de transmissão de dados, mas sim de acrescentar filtros e amplificadores.

Para amenizar esta inveja estamos publicando um "paliativo": o **PENTASPEED.** Como o nome sugere, o **PENTASPEED** é uma rotina (em ASSEMBLY) que permite ler e gravar programas com uma velocidade *cinco* vezes maior que o usual.

Esta velocidade é suficiente alta para valer a pena o trabalho de inserir a rotina no computador com 16K, mas suficientemente baixa para não exigir nada além de um gravador razoável e fita de boa qualidade.

Inicialmente digitamos o programa A da fig. 1, tomando o cuidado de inserir pelo menos 282 "zeros" na linha **1 REM**.

```
PROGRAMA A
   1 REM 00000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
000
  10 PRINT "ENDERECO INICIAL?"
  20 INPUT E
  30 PRINT E
  40 FOR I=E TO 16757
  50 SCROLL
  60 PRINT I;"...";
  70 INPUT P
  80 PRINT P
  90 POKE I,P
 100 NEXT I
 110 SCROLL
 120 PRINT "FIM"
```

A velocidade deste programa não é compatível com o **HIGH-SPEED**, não permitindo aproveitamento de fitas gravadas nesta velocidade. Em compensação ele pode ser carregado num TK85, que passará a ter 3 opções de velocidade de transmissão de dados.

Ao rodar este programa ele pede o endereço inicial. Devemos digitar:

16514

a seguir toda a lista de códigos decimais da fig. 2. O programa, através do **POKE** da linha 9Ø irá substituindo os "Ø" da linha **REM** pelos bytes da rotina em linguagem de máquina.

Se você errar algum byte não tem problema: digite **STOP** e **NEW LINE** interrompendo o programa. A seguir comande **RUN** e indique como endereço inicial o último em que você digitou corretamente.

```
LISTAGEM DOS ENDERECOS E CODIGOS
DECIMAIS DA ROTINA PENTASPEED
PARA INSERIR NA LINHA 1 REM

. . . . . . . . . . . . . . . . . . . . . . . . . .

   16514...205        16515...35
   16516...15         16517...17
   16518...6          16519...127
   16520...205        16521...43
   16522...127        16523...205
   16524...43         16525...15
   16526...201        16527...205
   16528...35         16529...15
   16530...33         16531...29
   16532...127        16533...34
   16534...22         16535...64
   16536...205        16537...112
   16538...127        16539...205
   16540...43         16541...15
   16542...201        16543...11
   16544...11         16545...11
   16546...0          16547...0
   16548...0          16549...0
   16550...0          16551...0
   16552...205        16553...168
   16554...3          16555...56
   16556...249        16557...235
   16558...17         16559...203
   16560...18         16561...205
   16562...70         16563...15
   16564...48         16565...46
   16566...16         16567...254
   16568...27         16569...122
   16570...179        16571...32
   16572...244        16573...205
   16574...78         16575...127
   16576...203        16577...126
   16578...35         16579...40
   16580...248        16581...33
   16582...9          16583...64
   16584...205        16585...78
   16586...127        16587...205
```

fig. 2

continua

```
16588...252
16590...24
16592...94
16594...203
16596...200
16598...230
16600...198
16602...79
16604...255
16606...35
16608...254
16610...70
16612...48
16614...6
16616...16
16618...13
16620...238
16622...216
16624...24
16626...205
16628...3
16630...18
16632...10
16634...124
16636...24
16638...14
16640...6
16642...62
16644...219
16646...211
16648...31
16650...73
16652...23
16654...40
16656...241
16658...186
16660...229
16662...98
16664...205
16666...127
16668...122
16670...32
16672...190
16674...214
16676...23
16678...241
16680...52
16682...33
16684...64
16686...205
16688...127
16690...205
16692...1
16694...246
16696...30
16698...6
16700...29
16702...254
16704...203
16706...123
16708...245
16710...245
16712...32
16714...254
16716...48
16718...63
16720...17
16722...173
16724...122
16726...40
16728...207
16730...167
16732...80
16734...254
16736...110
16738...33
16740...64
16742...0
16744...1
16746...0
16748...176
16750...255
16752...34
16754...64
16756...195
```

```
16589...1
16591...248
16593...55
16595...19
16597...159
16599...2
16601...1
16603...211
16605...6
16607...16
16609...205
16611...15
16613...114
16615...30
16617...254
16619...32
16621...195
16623...127
16625...224
16627...168
16629...203
16631...203
16633...205
16635...127
16637...251
16639...1
16641...0
16643...127
16645...254
16647...255
16649...48
16651...23
16653...56
16655...16
16657...241
16659...210
16661...3
16663...107
16665...124
16667...203
16669...121
16671...3
16673...32
16675...35
16677...48
16679...253
16681...21
16683...9
16685...80
16687...124
16689...113
16691...252
16693...24
16695...213
16697...49
16699...14
16701...219
16703...23
16705...123
16707...55
16709...16
16711...209
16713...4
16715...86
16717...178
16719...203
16721...48
16723...201
16725...167
16727...187
16729...12
16731...6
16733...16
16735...195
16737...127
16739...130
16741...17
16743...127
16745...224
16747...237
16749...33
16751...126
16753...4
16755...195
16757...3
```

Ao terminar esta tarefa apague todas as linhas do programa A, menos a 1 **REM**, agora toda cheia de uma estranha sequência de caracteres. A seguir digite as linhas de 1Ø a 7Ø do programa B (fig. 3).

```
PROGRAMA B

   1 REM LN 7?)▒LN F▒LN F?TAN L
N 7?51▒6-RNDLN ?▒LN F?TAN """
  LN ▒5 RAND FOR )ACS >LN ??KI
( RETURN .?▒4 POKE LN ?▒ACS RNDL
N ?▒LN UNPLOT / SAVE ?RACS (COS
▒ NEW LEN ?PEEK COPY 7( RETU
RN LN ??K?2( RETURN $4 INPUT ?*
*▒/ STEP LN ▒ACS >ACS LN ?▒/ C
LS : Y▒<= RETURN PEEK COPY 3K?
**SC( LET LET ▒ABS FAST ??LN ?▒
ACS ??4▒4CHR$ 7*K LET CLEAR 0+5
RND?LN ?▒?LN UNPLOT / PLOT STR
$ 2L:1<= RETURN *ACS ??S PRINT
( PRINT SGN 4 RETURN ?K▒ZACS )K
▒TAN ?▒C▒INT £▒?( RETURN ??▒5▒R
ND) ▒ STEP GOSUB ▒5 COPY 0000
00000000000000000000000000000000
00
  10 SAVE "PENTASPEED"
  20 PRINT TAB 8;"PENTASPEED - TK82/83"
  30 PRINT ,,,,,,"P/ GRAVAR DIGI
TE RAND USR 32512"
  40 PRINT "P/CARREGAR DIGITE RA
ND USR 32525"
  50 PRINT "PARA GUARDAR ESTA RO
TINA NO FIM  DA RAM DIGITE QUALQ
UER LETRA"
  60 PAUSE 40000
  70 RAND USR 16738
```

Prepare uma boa fita, coloque o gravador na posição gravar e comande **RUN**. O programa B vai se gravar (devido à linha 1Ø) e a seguir já se inicia (fig. 4).

```
        PENTASPEED - TK82/83


P/ GRAVAR DIGITE RAND USR 32512
P/CARREGAR DIGITE RAND USR 32512
PARA GUARDAR ESTA ROTINA NO FIM
DA RAM DIGITE QUALQUER LETRA
```

Ao digitar qualquer letra o programa abaixa o valor da **RAMTOP** de uns 250 bytes e transfere para esta região reservada o código de máquina contido na linha **1 REM**. A seguir o programa se autodestrói, e o computador fica com a RAM aparentemente vazia. Não se assuste: a rotina está escondidinha no topo da RAM e a única maneira de tirá-la de lá é desligar o computador. Pegue uma qualquer de suas fitas já gravadas e passe um programa para o computador. A seguir coloque uma fita virgem no gravador, digite:

RAND USR 32512

Coloque a fita rodando com o gravador em **REC** e digite **NEW LINE**. Seu programa está sendo passado para a fita em **PENTASPEED.**

Rebobine a fita, digite:

RAND USR 32525

e **NEW LINE**.

Ao passar da fita para o computador um programa de 5 minutos demorará apenas 1 minuto.

ENTRE O MICRO E O CASSETE:
CMS/ZX um bom sinal
GRAVADOR
CMS/ZX
MICRO
V.U.
55mm
Calibração Ótima
90°
45°
15mm
80mm
60mm
PREÇO DE LANÇAMENTO Cr$ 16.950,00
CMS/ZX: CALIBRADOR E MODULADOR DE SINAL
O microcomputador, criado para atender às necessidades pessoais, ou mesmo às pequenas e médias empresas, utiliza-se de um sistema simples e barato de armazenamento de dados: o gravador comum e a fita cassete.
O gravador, no entanto, ao transmitir o programa para o micro, nem sempre o faz de maneira a facilitar a leitura, por desajuste de sinal. Conseqüentemente, a transposição do programa na maioria das vezes, exige do usuário, várias tentativas para efetuar a operação. Por exemplo, um programa que passaria sem problemas em cinco minutos, chega a levar vinte minutos ou mais em leituras frustradas.
A POLIMICRO, para solucionar este problema, criou o CMS/ZX um calibrador e modulador de sinal de leitura de fitas magnéticas para microcomputadores de lógica Sinclair, ou seja o ZX 81, o NEZ 8000, o TK 82C e o CP200.
CMS/ZX
O CMS/ZX, modulando o sinal e informando através de um V.U. se o seu programa está sendo adequadamente transmitido, torna fácil a leitura do microcomputador.
O V.U. mede a intensidade do som, sendo ainda eficiente na monitorização em casos de gravação de programas e dados. A junção do CMS/ZX ao sistema é feito por dois fios, um ligado ao micro e outro ao cassete.
O CMS/ZX, aparelho de tecnologia e desenvolvimento inteiramente nacional, traz um bom sinal ao seu micro.
POLIMICRO avenida andrade neves, 1254 - campinas - fone (0192) 80822
Vendas através de reembolso postal ou revendedores de sua cidade
Credenciamos revendedores em todo o Brasil

CURSO DE BASIC TK

aula 4

Até agora, aprendemos a programar o computador para que ele executasse várias ordens seguidas; entretanto, durante a execução, ele só escrevia coisas na tela, fazia cálculos ou "buscava" coisas em suas memórias; em outras palavras, ele apenas "falava" mas sem "escutar"! É como, por exemplo, dizer a uma pessoa: (programá-la): fale os 10 primeiros números pares! Durante a "execução" da ordem, ela não necessitará nos ouvir, apenas irá dizer os números. Agora experimente dizer a essa pessoa: eu quero que você some 2 números! Ora, o que ela irá fazer? Ela terá que PERGUNTAR quais são os números e "parar sua execução" até que você diga a ela quais são os mesmos! Analogamente, no computador, existe uma instrução que faz com que, durante a execução, ele PARE e espere que você coloque um número através do teclado! Esta instrução está na tecla I: **INPUT.** Vamos então fazer com que o computador nos pergunte 2 números e a seguir os some:

```
10 INPUT A
20 INPUT B
30 PRINT A+B
```

Execute o programa. A tela fica em branco apenas com o cursor **L** ; não entre em pânico. Ele está obedecendo suas ordens. A primeira instrução diz para ele PARAR e esperar que alguém digite um número para a seguir colocá-lo na variável A na memória. Coloque então o número 50 e a seguir **NEW LINE**. Novamente a tela fica em branco; ora, ele está agora esperando o segundo valor. Digite 20 e **NEW LINE;** finalmente obteremos a soma. . . Note entretanto que a tela em branco é um inconveniente que pode confundir bastante, principalmente se o programa for razoavelmente complicado. Portanto, é útil escrever algumas "mensagens", principalmente se o seu programa irá ser usado por alguma outra pessoa; essas mensagens NÃO interferem na lógica do programa, apenas ajudam a sua INTERPRETAÇÃO durante a EXECUÇÃO do mesmo; experimente então o seguinte:

```
   4 PRINT "EU VOU SOMAR 2 NUMER
OS"
   8 PRINT "QUAL O PRIMEIRO ?"
  10 INPUT A
  18 PRINT "QUAL O SEGUNDO ?"
  20 INPUT B
  25 PRINT "A SOMA E▪ ";
  30 PRINT A+B
```

Execute o programa. Note que as linhas 4, 8, 18 e 25 são apenas MENSAGENS. Acrescente então as seguintes linhas:

```
 6 PRINT
12 PRINT A
14 PRINT
21 PRINT B
22 PRINT
```

Supondo, por exemplo, que você entre com os números 60 e 40 teremos na tela:

```
EU VOU SOMAR 2 NUMEROS

QUAL O PRIMEIRO ?
60

QUAL O SEGUNDO ?
40

A SOMA E▪ 100
```

Como você faria para os números 60 e 40 aparecerem não em baixo da pergunta mas IMEDIATAMENTE ao lado da mesma? E para que eles aparecessem na mesma linha da pergunta mas na METADE da tela? (Lembre-se do efeito do ";" e da ","; basta colocá-los convenientemente nas instruções 8 e 18).

Tenha então sempre em mente que, ao programar um computador, 3 cérebros estão em jogo: o seu, o do com-

putador e o da pessoa que irá usar o programa. É por isto que as "mensagens" são fundamentais.

Vamos agora apresentar outra KEY-WORD: trata-se do CLS (tecla V); acrescente a linha:

```
23 CLS
```

O que acontece? Esta instrução APAGA tudo o que havia sido escrito **na tela** até então. Note entretanto que fica difícil ver o efeito da instrução PRINT B pois apagamos a tela logo em seguida. Para evitar isto, seria necessário uma "espera" ou pausa antes que ele executasse essa instrução; acrescente então a linha:

```
24 PAUSE 120                 (LETRA M)
```

e veja o que ocorre. Esta instrução acrescenta uma PAUSA ao programa onde para cada SEGUNDO de espera devemos adicionar 6Ø; no caso, a pausa é de 2s; se eu quizesse uma pausa de 3.5s deveria colocar PAUSE 21Ø. Note que durante um INPUT quando o computador está "parado" esperando um número, não adianta fazer BREAK para apagar o programa, pois ele já está parado; assim, se você quer interromper, deverá pressionar STOP (SHIFT e tecla A) e a seguir NEW LINE.

Cabem aqui algumas informações quanto à precisão do computador; ele só reconhece 8 dígitos diferentes de zero. De fato, usando o programa anterior, tente somar 111111111 com Ø; você verá que é apresentado um resultado ERRADO pois estamos acima da precisão para este tipo de NOTAÇÃO de números; note entretanto que se o resultado tiver **zeros** no final ele é capaz de representar até a 13ª casa; some 999.999.999.999 com 1 e o resultado estará correto. Se quisermos números maiores, outra notação terá que ser utilizada; tente somar: 9.999.999.999 com 1; você obterá 1E + 13 que significa 1×10^{13}; assim o computador é capaz de interpretar números com mais de 8 dígitos desde que usemos esta notação. O mesmo é válido para números muito pequenos. O maior número em valor absoluto que o computador TK aceita é:

1.7Ø14118E + 38

e o menor em valor absoluto (o mais próximo de zero):

2.9387359E − 39

Vamos fazer um programa que utilize as instruções aqui apresentadas, e mais uma função que está na tecla G (ABS) que serve para calcular o valor ABSOLUTO de um número, ou seja, seu módulo; para entendê-la, experimente antes fazer este pequeno programa:

```
10 FOR I=10 TO -10 STEP -1
20 PRINT TAB (ABS I);I
30 NEXT I
```

Que tal o efeito?

Feito isto, iremos então fazer um programa para "treinar a tabuada", que pergunte para você o resultado de uma operação de multiplicação usando números "centrados" por ele escrevendo a sua resposta, a resposta do computador e a diferença entre elas em valor absoluto. Vamos fazer com que ele pergunte também qual o valor máximo dos números que ele deve multiplicar:

```
  10 RAND
  20 PRINT "EU VOU PERGUNTAR A V
OCE QUANTO  VALE O PRODUTO DE 2
NUMEROS"
  30 PRINT "QUAL O NUMERO MAXIMO
QUE EU POSSO USAR ?"
  40 INPUT MAX
  50 CLS
  60 LET A=INT (2*MAX*RND+1)-MAX
  70 LET B=INT (2*MAX*RND+1)-MAX
  80 PRINT A;"X";B;"=?"
  90 INPUT P
 100 PRINT
 110 PRINT "SUA RESPOSTA ";P
 120 PRINT "RESPOSTA CORRETA ";A
*B
 130 PRINT
 140 PRINT "DIFERENCA ";ABS (P-A
*B)
 150 PAUSE 180
 160 CLS
 170 GOTO 20
```

Note que na linha 17Ø foi usada uma nova KEY-WORD: GOTO (tecla G) que significa "vá para" dada instrução (no caso 2Ø); seu efeito é análogo ao RUN só que ela NÃO executa um CLEAR. De fato, pare o programa (usando BREAK ou STOP conforme o caso) e, execute-o novamente; só que, ao invés de fazer RUN e NEW-LINE faça GOTO 1Ø e NEW LINE!

Preste atenção na estrutura utilizada nas linhas 6Ø e 7Ø para "gerar" números positivos ou negativos com valor absoluto menor do que o valor máximo! O

QUEBRA-CABEÇA

Este mês:

AS IDADES DAS TRÊS FILHAS

O quebra-cabeça deste mês consiste em fazer com que, de algum modo, o TK resolva um problema já bastante conhecido.

Um matemático (A), propôs a um de seus amigos, também matemático (B), que ele "adivinhasse" as idades de suas três filhas. Entre eles transcorreu o seguinte diálogo.

Matemático A: "O produto das idades de minhas três filhas é igual a 36".

Matemático B: "Só com essa informação é impossível saber quais as três idades".

Matemático A: "Está bem. Saiba, então que a soma das idades das minhas filhas é igual ao número daquela residência" (Disse isso apontando para uma casa próxima de onde se encontravam).

Matemático B: "Ainda assim, me é impossível precisar as idades".

Matemático A: "Vou lhe dar uma última pista. A mais velha de minhas filhas dança "ballet".

O outro matemático pensou durante algum tempo e finalmente afirmou:

Matemático B: "Com certeza, as idades de suas filhas são . . . "

Obviamente, as idades devem ser consideradas como números inteiros e em anos. Assim, uma pessoa com 18 anos e 6 meses de vida tem idade igual a 18 anos.

A exemplo do Quebra-Cabeça do nº 2 (Verdadeiro ou Falso), não estamos interessados nas três idades das filhas do Matemático (A), mas sim num programa que as obtenha. Aqui, também, é conveniente utilizar preferencialmente as instruções e comandos destinados à análises lógicas:

(IF . . . THEN, OR, AND, NOT, > –, < =, etc.).

○

V ou F? solução

Renato da Silva Oliveira

Para um ser humano, talvez a forma mais fácil de resolver esse problema seja escolher entre as dez (1Ø) asserções as duas mais antagônicas, (isto é, as duas cujos valores V ou F, sejam necessariamente diferentes) e atribuir a uma delas um valor. A partir daí, obter os valores das demais asserções e verificar se há coerência entre elas. Passando as vistas pelas dez afirmações a dupla escolhida provavelmente será formada pelas asserções 9 e 1Ø. De fato, se a asserção 9 for verdadeira (valor V) a asserção 1Ø será falsa (valor F); e, se a asserção 9 for falsa (valor F) a asserção 1Ø será verdadeira (valor V).

Para facilitar a análise das asserções, nós as designaremos convenientemente por A(1), A(2), . . . até A(1Ø) seguindo a mesma ordem em que elas são apresentadas.

Pode-se fazer, por exemplo:

A(1Ø) = F ; portanto . . .
A(9) = V ; portanto . . .

. . . e assim por diante, até concluir algo a respeito da coerência ou não dos valores dados à asserções.

No exemplo acima, em que supôs-se A(1Ø) = F, chega-se a uma incoerência, o que mostra que, **se há uma solução,** nela o valor de A(1Ø) tem que ser V. Tomando-se A(1Ø) = V obtém-se o resultado coerente mostrado abaixo:

A(1) = V	A(6) = F
A(2) = F	A(7) = F
A(3) = V	A(8) = V
A(4) = F	A(9) = F
A(5) = V	A(1Ø) = V

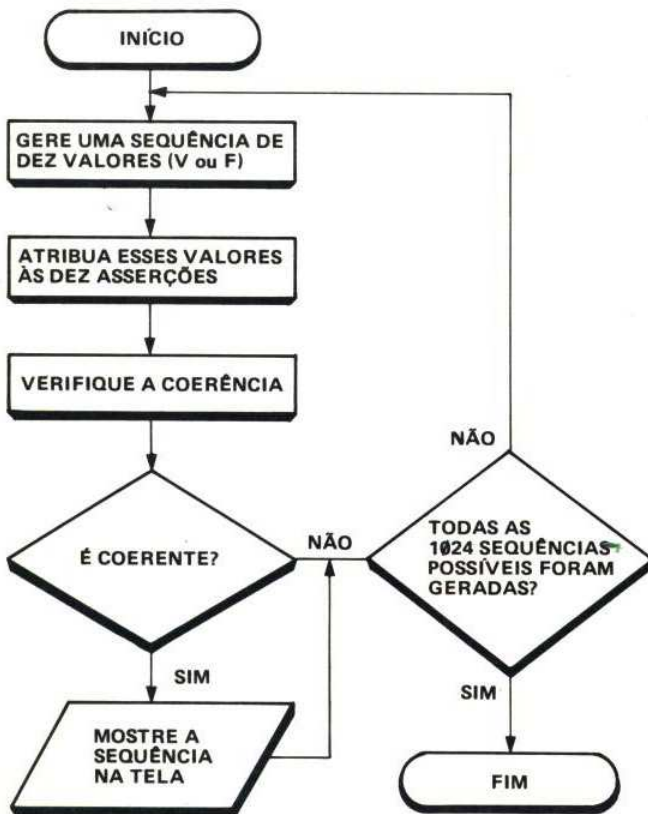
A solução do problema, entretanto, não é solução do "Quebra-Cabeça". O que queremos é um programa para que o TK encontre a solução!

Podemos pensar em fazer um programa que "instrua" o TK para que ele proceda de modo análogo ao de um ser humano, entretanto, além de extremamente complexo, um programa desse tipo deve ser deveras extenso e, por conseguinte, levará muito tempo para ser rodado completamente. Provavelmente, tanto para o "micro" como para seu programador, o processo mais fácil e rápido é, sem nenhum paradoxo, o mais simples e trabalhoso .

continua

(Por exemplo: Fazer mil contas de somar é mais simples e mais trabalhoso que fazer uma única extração de raíz quadrada).

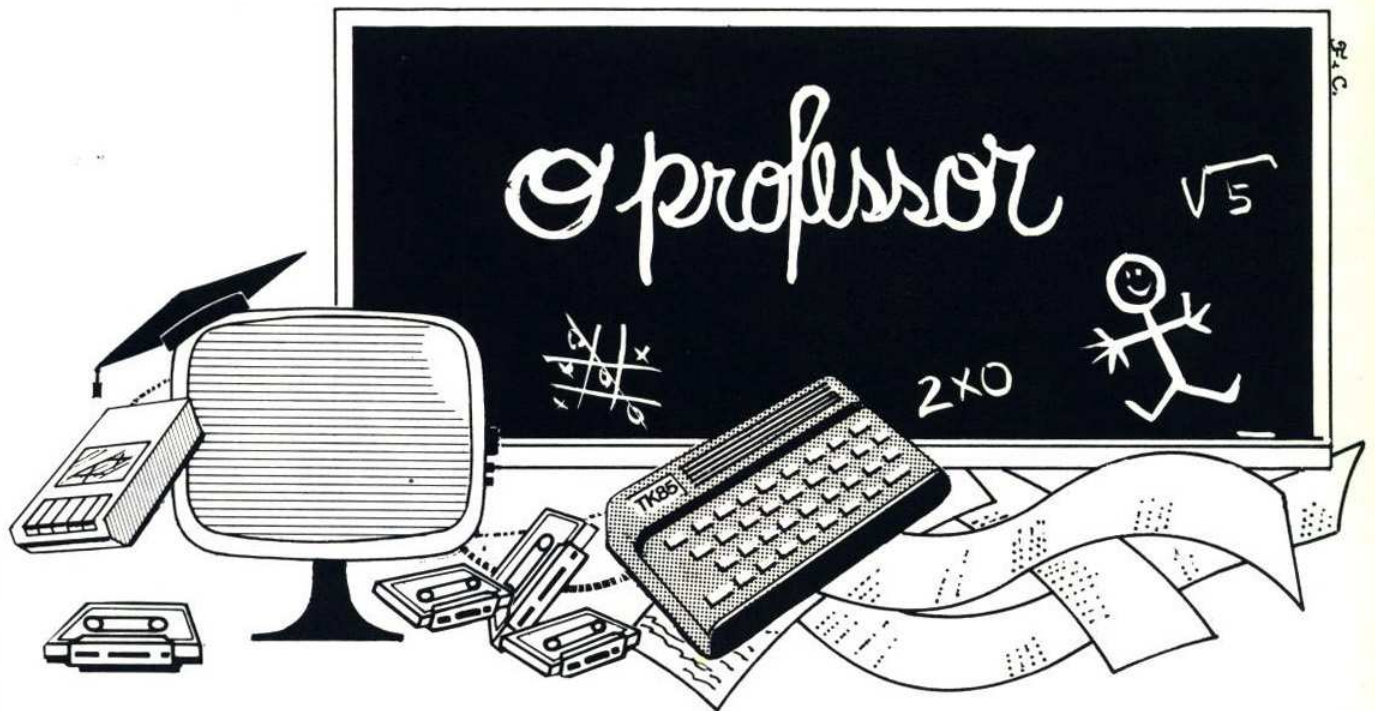
Tal processo consiste apenas em atribuir a cada asserção um valor (V ou F) e para cada sequência de valores dados a A(1), A(2), . . ., A(1Ø) verificar se há ou não coerência. As asserções são em número de dez (1Ø), cada uma podendo assumir dois valores (V ou F), portanto o número de sequências possíveis é 2^{10} = 1Ø24 (desde A(1) = A(2) = . . . = A(1Ø) = F até A(1) = A(2) = ... = A(1Ø) = V).

O que o TK terá que fazer é gerar cada uma das 1Ø24 sequências de valores para as asserções e testá-las uma por uma, mostrando na tela apenas as que forem coerentes e, portanto, soluções do problema. O fluxograma a ser seguido deve ser algo semelhante ao mostrado abaixo:

Ao tentar traduzir o fluxograma para a linguagem BASIC encontramos maior dificuldade quando temos que escrever o conteúdo das asserções numa linguagem conveniente para o TK. Uma solução possível é aproveitar as próprias características do micro e chamar de **1** (um) ao valor V e de **Ø** (zero) ao valor F, utilizando principalmente as instruções e comandos destinados a execução de operações lógicas (**IF . . . THEN; OR; AND; NOT; > =; < =; . . . etc . . .**).

Abaixo encontra-se listado um dos possíveis programas obtidos como sugerido pelo fluxograma. Digite-o e verifique você mesmo o resultado. Se você tem apenas 2 Kbytes de RAM, não se assuste com o fato de a listagem confinar-se ao tôpo da tela. Isso acontece por que esse programa ocupa 1688 bytes e não sobra lugar na memória para o restante da listagem ser mostrado na tela. O principal inconveniente desse programa é seu tempo de processamento (cerca de 25 min). Como sugestão, ponha-o para rodar e vá tomar um café e ler um pouco de jornal, superando engenhosamente o inconveniente!

```
  10 REM "V OU F"
  20 FAST
  30 LET D=VAL "7"
  40 DIM A(10)
  50 FOR N=PI/PI TO VAL "2**10"
  60 LET M=N-PI/PI
  70 FOR K=PI/PI TO VAL "10"
  80 LET B=M/2
  90 LET A(K)=M-2*INT (B)
 100 LET M=INT (B)
 110 NEXT K
 200 DIM S(10)
 210 FOR X=PI/PI TO VAL "10"
 220 LET O=PI-PI
 230 FOR L=PI/PI TO X
 240 LET O=O+A(L)
 250 NEXT L
 260 LET S(X)=O
 270 NEXT X
 300 DIM V(10)
 310 LET V(PI/PI)=((S(4)=PI/PI)
OR (S(6)=3) OR (S(8)=5))
 320 LET V(2)=(S(3)<(S(10)-S(6))
)
 330 LET V(3)=(S(10)<=5)
 340 LET V(4)=((S(3)=3) OR (S(4)
-S(1)=3) OR (S(5)-S(2)=3) OR (S(
6)-S(3)=3) OR (S(7)-S(4)=3) OR (
S(8)-S(5)=3) OR (S(9)-S(6)=3) OR
 (S(10)-S(7)=3))
 350 LET V(5)=(S(10)>=5)
 360 LET V(6)=(A(7)=A(10))
 370 LET V(7)=(A(6)+A(8)+A(10)=P
I/PI)
 380 IF A(10)=PI/PI THEN LET V(8
)=((A(6)=PI-PI) AND (A(9)=PI-PI)
)
 381 IF A(10)=PI-PI THEN LET V(8
)=PI/PI
 390 LET V(9)=(((S(2)=2) OR ((A(
2)+A(4)=PI-PI) AND (A(5)+A(6)=2)
) OR ((A(2)=PI-PI) AND (A(3)+A(4
)=2))) AND (A(6)=PI/PI))
 400 LET V(10)=(A(9)=PI-PI)
 410 FOR R=PI/PI TO VAL "10"
 420 IF V(R)=A(R) THEN GOTO 440
 430 GOTO 500
 440 NEXT R
 450 FOR C=PI/PI TO VAL "10"
 460 IF V(C)=PI/PI THEN PRINT AT
 C,D;"V"
 470 IF V(C)=PI-PI THEN PRINT AT
 C,D;"F"
 480 NEXT C
 490 LET D=D+VAL "2"
 500 NEXT N
 510 FOR Z=PI/PI TO VAL "9"
 520 PRINT AT Z,PI/PI;"A( ";Z;")
="
 521 NEXT Z
 525 PRINT AT VAL "10",PI/PI;"A(
10)="
```

As linhas de 5Ø a 11Ø geram uma sequência de 1Ø valores. As de 21Ø a 27Ø apenas auxiliam na análise da coerência. As linhas de 3ØØ a 4ØØ são as asserções traduzidas e escritas de forma conveniente (verifique como!) para o TK. As linhas de 41Ø a 44Ø testam a coerência e, finalmente, as linhas de 45Ø a 525 apresentam o(s) resultado(s)** na tela.

** (Nada garante que haja apenas uma solução a princípio!)

O Professor

Bernardo C. Stein

O seu computador **TK82** ou **TK85** pode ser usado como um eficiente professor. O Programa que apresentamos adiante foi criado para auxiliá-lo a memorizar pares de informações, como por exemplo, países e suas capitais, elementos químicos e seus símbolos, palavras em português e sua tradução em inglês, e assim sucessivamente.

Ele foi adaptado a partir de um programa publicado originalmente na revista americana SYNC de Set/out de 82.

Se você tiver três horas de paciência para entrar com o programa no seu TK, você será recompensado com um excelente auxiliar de instrução. Vale a pena. . . Vamos lá?

1ª Parte: ENTRADA DO PROGRAMA

Entre com o programa conforme listado na figura 1. É recomendável ir gravando o programa em fita à medida que você for digitando, para não correr o risco de perder todo o seu trabalho com alguma falta de energia elétrica, por exemplo.

```
   7 REM       O PROFESSOR
         TRADUZIDO E ADAPTADO
            POR B. C. STEIN
              JUNHO 1983

   8 REM PARA "RUN" O PROGRAMA,
ENTRE "GOTO 100"
   9 REM PARA ENTRAR COM NOVOS
DADOS OU PARA MANIPULAR O PROGRA
MA, ENTRE "GOTO 1000"
  10 LET B=INT (RND*TOT)+1
  12 CLS
  13 PRINT AT 4,0;"-DIGITE Q PAR
A RETORNAR AO MENU-"
  14 RETURN
  15 LET D$=INKEY$
  16 IF D$="Q" THEN GOTO 990
  17 RETURN
  20 PRINT "NESTA OPCAO, ";C$;"

  21 RETURN
  25 PRINT "VOCE DISSE QUE E ";
D$;"."
  26 RETURN
  30 PRINT TAB 8;"VOCE ACERTOU."
  31 RETURN
 100 PRINT AT 5,0;"O PROFESSOR D
E"
 102 PRINT T$
 104 PRINT AT 10,0;"POR FAVOR, E
NTRE COM O SEU NOME."
 110 INPUT C$
 115 CLS
 120 PRINT AT 4,0;"COMO VAI  ";C
$;"?"
 121 PRINT "EU VOU AJUDAR VOCE A
 CONHECER    ";T$;"."
 122 PRINT "DADO/A UM/A ";N$;","
 123 PRINT "VOCE IRA APRENDER SE
U/SUA       "
 124 PRINT N$;", E VICE-VERSA."
 126 PRINT
 127 SLOW
 128 PRINT TAB 10;"MENU"
 129 PRINT
 130 PRINT "   1. REVISAO DA LIST
A"
 131 PRINT "   2. TESTE PRATICO"
 132 PRINT "   3. TESTE DE RAPIDE
Z"
 133 PRINT "   4. FIM DA AULA"
 134 PRINT
 135 PRINT "ESCOLHA UMA DAS OPCO
ES ACIMA."
 140 INPUT A
 142 IF A<>1 AND A<>2 AND A<>3 A
ND A<>4 THEN GOTO 140
 145 CLS
 150 GOTO A*200
 200 PRINT AT 5,0;"--------REVIS
AO DA LISTA--------"
 205 GOSUB 20
 210 PRINT "EU VOU MOSTRAR UM/UM
A ";N$;" "
```

continua

```
 211 PRINT "E RESPECTIVO/A ";M$;
"."
 212 PRINT "DEPOIS DE MOSTRAR OS
";TOT;" ITENS,  PASSAREI A RELA
CAO AO CONTRARIO."
 213 PRINT " ESCOLHA UM NUMERO D
E 0 A 9,        0-PASSAGEM MAIS RAP
IDA,            9-PASSAGEM MAIS LEN
TA."
 215 INPUT Z
 217 IF Z<0 OR Z>9 THEN GOTO 215
 220 FOR A=1 TO TOT
 225 GOSUB 12
 230 PRINT AT 10,2;B$(A);"  ";
 235 FOR N=0 TO 4+2*Z
 236 NEXT N
 240 GOSUB 15
 245 PRINT A$(A)
 250 FOR N=0 TO 4+2*Z
 251 NEXT N
 255 GOSUB 15
 260 NEXT A
 265 FOR A=1 TO TOT
 270 GOSUB 12
 275 PRINT AT 10,2;A$(A);"  ";
 280 FOR N=0 TO 4+2*Z
 281 NEXT N
 285 GOSUB 15
 290 PRINT B$(A)
 295 FOR N=0 TO 4+2*Z
 296 NEXT N
 300 GOSUB 15
 305 NEXT A
 310 GOTO 220
 400 PRINT AT 5,0;"---------TEST
E  PRATICO---------"
 405 GOSUB 20
 410 PRINT "VOCE RESPONDERA A 30
 QUESTOES:","            QUESTAO 1
A 15"
 411 PRINT "EU LHE DOU UM/A ";M$
 412 PRINT "VOCE ENTRA COM SEU/S
UA ";N$;"."
 413 PRINT "             QUESTAO 16
A 30"
 414 PRINT "FAREMOS O INVERSO."
 415 PRINT "ENTRE ""NEW LINE"" P
ARA COMECAR"
 417 LET U=0
 418 INPUT Z$
 419 IF F2 THEN GOTO 435
 420 FOR A=1 TO TOT
 425 LET E$(A)=CHR$ 0
 430 NEXT A
 435 FOR A=1 TO 15
 440 GOSUB 10
 445 IF E$(B)="X" THEN GOTO 440
 447 PRINT AT 9,0;"QUAL E O/A ";
N$;" DO/A"
 450 PRINT AT 10,8;A$(B)
 460 INPUT D$
 462 GOSUB 25
 465 GOSUB 16
 470 IF LEN D$=L2 THEN GOTO 475
 472 LET D$=D$+" "
 473 GOTO 470
 475 IF D$=B$(B) THEN GOTO 490
 476 PRINT "VOCE ERROU, O CERTO
E "
 477 PRINT TAB 8;B$(B)
 478 FOR N=0 TO 35
 479 NEXT N
 480 GOTO 497
 490 IF NOT F2 THEN LET E$(B)="X
"
 492 GOSUB 30
 493 LET U=U+1
 495 FOR N=0 TO 16
 496 NEXT N
 497 NEXT A
 500 FOR A=1 TO 15
 505 GOSUB 10
 510 IF E$(B)="X" THEN GOTO 505
 512 PRINT AT 9,0;"QUAL E O/A ";
M$;","
 513 PRINT "CUJO/A ";N$;" E,"
```

```
 515 PRINT AT 11,8;B$(B)
 520 INPUT D$
 525 GOSUB 25
 527 GOSUB 16
 530 IF LEN D$=L1 THEN GOTO 545
 535 LET D$=D$+" "
 540 GOTO 530
 545 IF D$=A$(B) THEN GOTO 560
 550 PRINT "VOCE ERROU, O CERTO
E "
 551 PRINT TAB 8;A$(B)
 553 FOR N=0 TO 40
 554 NEXT N
 555 GOTO 575
 560 GOSUB 30
 565 IF NOT F2 THEN LET E$(B)="X
"
 570 FOR N=0 TO 20
 571 NEXT N
 572 LET U=U+1
 575 NEXT A
 580 CLS
 584 SLOW
 585 PRINT AT 9,0;C$;", VOCE FEZ
 ";U;" PONTOS."
 586 PRINT
 590 PRINT "DIGITE ""NEW LINE""
PARA O MENU."
 592 INPUT D$
 595 GOTO 990
 600 PRINT AT 5,0;"--------TESTE
DE RAPIDEZ--------"
 605 GOSUB 20
 610 PRINT "VOCE TEM 60 SEGUNDOS
PARA ENTRARCOM O MAIOR NUMERO P
OSSIVEL DE  RESPOSTAS."
 611 PRINT "EU DOU UM/A ";M$;","

 612 PRINT "VOCE ENTRA SEU/SUA "
;N$;"."
 613 PRINT
```

continua

```
 614 PRINT "  +1 PONTO POR RESPOS
TA CERTA,     -1 PONTO POR RESPOS
TA ERRADA,     0 PONTOS SE VOCE N
AO SOUBER        E ENTRAR ""NEW L
INE""."
 615 PRINT
 616 PRINT "ENTRE ""NEW LINE"" P
ARA COMECAR"
 617 LET U=0
 618 FAST
 619 INPUT D$
 620 POKE 16436,0
 625 POKE 16437,0
 630 GOSUB 10
 640 PRINT AT 10,8;A$(B)
 645 INPUT D$
 647 IF PEEK 16436+256*PEEK 1643
7<62500 THEN GOTO 580
 650 IF D$="" THEN GOTO 630
 655 GOSUB 16
 657 IF LEN D$=L2 THEN GOTO 660
 658 LET D$=D$+" "
 659 GOTO 657
 660 IF D$<>B$(B) THEN LET U=U-1
 665 IF D$=B$(B) THEN LET U=U+1
 670 GOTO 630
 800 PRINT AT 9,0;"VOCE SE SAIU
MUITO BEM,        ";C$;"."
 805 FOR N=0 TO 20
 806 NEXT N
 807 PRINT
 810 PRINT "ESPERO QUE TENHA GOS
TADO DE      APRENDER ALGO NOVO."
 812 PRINT
 815 FOR N=0 TO 20
 816 NEXT N
 820 PRINT "*******ATE* A PROXIM
A VEZ*******"
 822 FOR N=0 TO 200
 823 NEXT N
 824 CLS
 825 GOTO 100
 990 CLS
 991 PRINT AT 5,14;""
 995 GOTO 126
 996 SAVE T$
 997 GOTO 100
 999 CLS
1000 PRINT AT 6,0;"ROTINA PARA C
RIACAO E REVISAO    DE DADOS."
1005 GOTO 1328
1008 LET F1=0
1009 PRINT AT 10,5;"ENTRE COM O
TITULO DO                SEU PROG
RAMA."
1010 INPUT T$
1012 LET F2=0
1015 CLS
1020 PRINT AT 9,0;"ENTRE COM O T
ITULO (NO SINGULAR)DO PRIMEIRO G
RUPO DE ITENS DE    SUA  RELACAO,
 DE PREFERENCIA OS ITENS MAIS LO
NGOS."
1030 INPUT M$
1035 CLS
1040 PRINT AT 9,0;"QUANTOS DIGIT
OS TERA A PALAVRA   MAIS LONGA DO
 PRIMEIRO GRUPO DE ITENS?"
1050 INPUT L1
1055 CLS
1060 PRINT AT 9,0;"ENTRE COM O T
ITULO (NO SINGULAR)DO SEGUNDO GR
UPO DE ITENS."
1070 INPUT N$
1073 CLS
1075 PRINT AT 9,0;"QUANTOS DIGIT
OS TERA A PALAVRA   MAIS LONGA DO
 SEGUNDO GRUPO DE   ITENS?"
1080 INPUT L2
1085 CLS
1090 PRINT AT 9,0;"QUANTOS ITENS
, NO TOTAL, TERA A SUA RELACAO?"
1095 INPUT TOT
1100 CLS
1105 PRINT AT 5,0;"TITULO-"
1106 PRINT T$
1110 PRINT "TITULO DO 1. GRUPO D
E ITENS:-   ";M$
1115 PRINT "MAIOR PALAVRA DO 1.
GRUPO:-       ";L1;" LETRAS."
1120 PRINT "TITULO DO 2. GRUPO D
E ITENS:-   ";N$
1125 PRINT "MAIOR PALAVRA DO 2.
GRUPO:-       ";L2;" LETRAS."
1130 PRINT "TOTAL DE ITENS DA RE
LACAO:-       ";TOT
1135 PRINT
1140 PRINT "AS INFORMACOES ESTAO
 CERTAS-S/N?"
1142 INPUT Z$
1145 IF Z$="S" THEN GOTO 1151
1147 IF Z$="N" THEN RUN 999
1150 IF Z$<>"S" OR Z$<>"N" THEN
GOTO 1140
1151 IF TOT<40 THEN LET F2=1
1160 DIM A$(TOT,L1)
1165 DIM B$(TOT,L2)
1170 DIM E$(TOT)
1175 CLS
1185 PRINT AT 10,0;"ENTRE COM O
1. ITEM DO 1. GRUPO,DEPOIS COM O
 1. ITEM DO SEGUNDO GRUPO; ENTRE
 COM O 2. ITEM DO 1.GRUPO, O 2.
ITEM DO 2. GRUPO,           E ASS
IM POR DIANTE."
1190 FOR A=1 TO TOT
1200 IF NOT F1 THEN GOTO 1250
1205 SCROLL
1210 PRINT M$;"--";A$(A)
1215 SCROLL
1220 PRINT N$;"--";B$(A)
1225 INPUT Z$
1230 IF Z$="" THEN GOTO 1290
1250 SCROLL
1255 PRINT M$;"--";
1260 INPUT A$(A)
1265 PRINT A$(A)
1270 SCROLL
1275 PRINT N$;"--";
1280 INPUT B$(A)
1285 PRINT B$(A)
1290 NEXT A
1292 CLS
1295 IF F1 THEN GOTO 1320
1300 PRINT AT 7,0;"   NOS VAMOS A
GORA REVER CADA PAR   DE ITENS.
                      QUANDO ELES
 APARECEREM NA TELA   ENTRE ""NEW
 LINE"",SE ESTIVEREM    CORRETOS.
"
1301 PRINT "  ENTRE ""X"", E ""N
EW LINE"", SE ES  TIVEREM ERRADO
S."
1302 PRINT "  NESTE CASO, VOLTE
A ENTRAR COM   O ITEM."
1305 LET F1=1
1310 GOTO 1190
1320 PRINT AT 5,5;"VOCE PODE AGO
RA:"
1328 PRINT
1330 PRINT
1335 PRINT "1. CRIAR UM NOVO ARQ
UIVO."
1336 PRINT "2. REFAZER A REVISAO
 DO ARQUIVO."
1338 PRINT "3. ""SAVE"" O PROGRA
MA."
1339 PRINT
1340 PRINT "ESCOLHA UMA DAS OPCO
ES."
1345 INPUT A
1346 CLS
1347 IF A=3 THEN GOTO 1370
1348 IF A=1 THEN RUN 1007
1350 IF A=2 THEN GOTO 1308
1355 GOTO 1330
1360 CLS
1370 PRINT AT 10,0;" ACIONE SEU
 GRAVADOR NA POSICAO  ""REC"", E
ENTRE ""NEW LINE""."
1380 INPUT Z$
1385 CLS
1390 IF Z$="" THEN GOTO 996
1400 STOP
```

continua

Uma vez digitado o programa proceda da seguinte forma:

2ª Parte: MANIPULAÇÃO DO PROGRAMA

1) Se você estiver no modo **FAST,** volte ao modo **SLOW**

2) Nunca, nunca mesmo, entre com **RUN**

3) Digite **GOTO 1ØØØ**

4) Aparecerá o "MENU" com 3 opções (veja fig. nº 2)

5) Escolha a opção nº 1 e **NEW LINE.**

6) Siga as instruções do programa:

a — escolha o título; por exemplo, PAÍSES E SUAS CAPITAIS (este será também o título com o qual o programa será gravado mais adiante.

b — escolha título (no singular) da 1ª série de dados: PAÍS (no nosso exemplo).

c — forneça o número de letras da palavra mais comprida, por exemplo: 15.

d — e — siga o mesmo procedimento para o segundo grupo de informações. Por exemplo: CAPITAL e 15 dígitos.

f — Entre agora com o número de países e capitais que você quer listar, por exemplo, 60.

7) O programa então lhe auxiliará a fazer uma revisão dos dados entrados até agora. Se houver algum erro entre **"N"** (não) e volte a fornecer os dados solicitados.

8) Se estiver tudo certo entre "**S**" (sim) e, então, comece a fornecer ao computador os dados (no nosso exemplo: país e sua respectiva capital) até completar a série.

Se você perceber que cometeu alguns erros, prossiga: haverá oportunidade de se fazer a correção por ocasião da revisão.

Após entrar com todos os pares de informações o programa auxiliará você a fazer uma revisão. A cada par de dados entre "**NEW LINE**" se a informação estiver correta. Entre "**X**" se estiver errado e corrija a informação, entrando novamente com o país e sua capital.

9) Após a revisão, o programa volta ao menu. Neste ponto escolha a opção 3. Coloque seu gravador em "REC" e grave (**SAVE**) o programa em fita.

3ª parte — USO DO PROGRAMA

Depois de todo nosso trabalho e dedicação (ufa. . .), estamos agora prontos para usufruir das aulas do nosso professor cibernético.

1) Após ter gravado o programa em fita ou quando, numa outra oportunidade, você passá-lo da fita para o gravador, usando o comando **LOAD** "PAÍSES E SUAS CAPITAIS" (no nosso exemplo, é claro) ele se iniciará automaticamente convidando você a fornecer seu nome.

2) A partir desse momento o tratamento entre o computador e você será personalizado, ele se dirigirá à você e lhe mandará o menu de opções que você tem para memorizar os "PAÍSES E SUAS CAPITAIS" (veja fig. 3).

3) Escolhendo a opção nº 1: revisão da lista, você poderá rever as capitais e seus respectivos países e vice-versa.

Você pode escolher a velocidade de revisão sendo Ø a mais rápida e 9 a mais lenta.

Para interromper a revisão entre **Q** e **NEW LINE** que também interromperá as opções nº 2 e 3 e retornará ao MENU.

4) Na opção nº 2:

O computador escolherá aleatóriamente 15 países e você deve entrar com a sua respectiva capital.

Depois ele escolherá 15 capitais e você deverá fornecer o respectivo país. No fim você será informado quantas respostas corretas você deu. No caso de você ter mais do que 40 ítens fornecidas ao computador ele não repetirá a pergunta cuja resposta dada estiver correta.

Na opção 3, que é um teste de rapidez, você deverá fornecer em segundos o maior número possível de capitais para os países perguntados. Cada resposta certa vale 1 ponto, cada resposta errada você perde um ponto. Se você só entrar **NEW LINE** você não perde nem ganha.

A opção nº 4 é um final para o programa, que se reiniciará em segundos. Nesta opção é fácil dar um "**BREAK**" no programa para poder digitar "**GOTO 1ØØØ**" e preparar uma aula diferente.

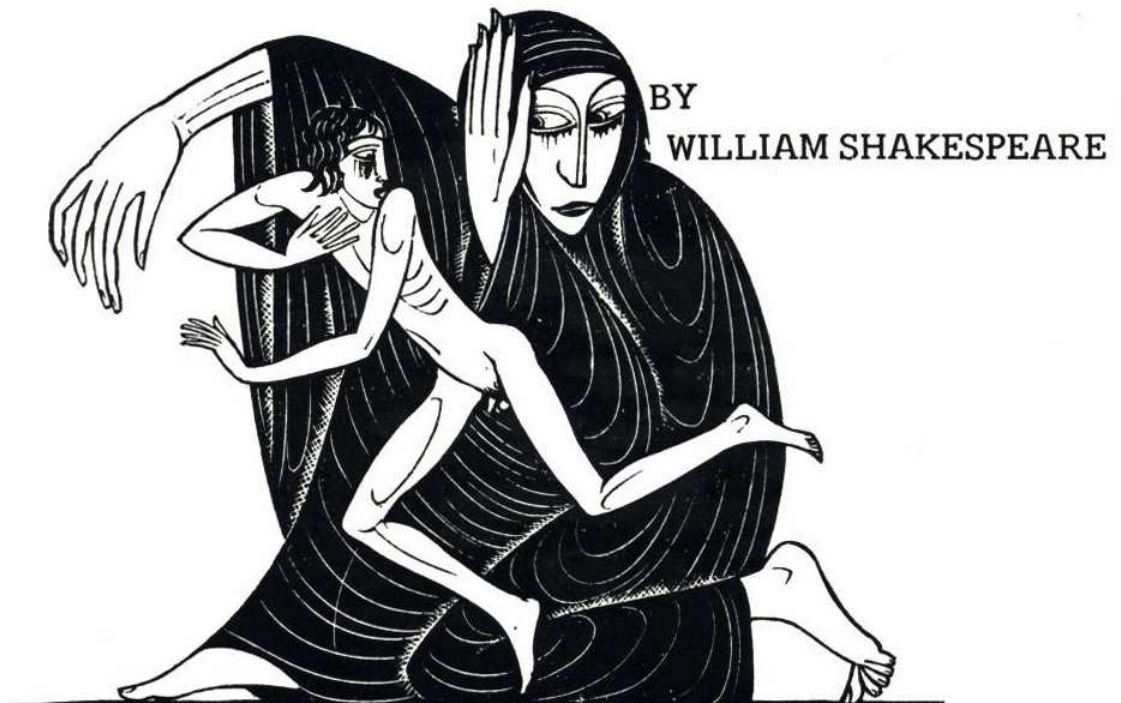
Claro que você pode preparar diversas aulas "diferentes" com o PROFESSOR e armazenar cada uma, com seu respectivo nome, em fita.

Se você quiser armazenar o programa em fita sem aula alguma, sugerimos que após ter entrado com ele no computador, armazená-lo em fita digitando **SAVE** "O PROFESSOR".

Quando, em outra ocasião, você quiser usá-lo para preparar uma aula entre **LOAD** "O PROFESSOR" e após tê-lo no computador entre "**GOTO 1ØØØ**".

Boa aula e boas notas!

PROGRAMA DO LEITOR

OTHELLO

Edson Mikio Yoshida
Nelson Murassaki

Este programa foi originalmente publicado na revista japonesa "**BASIC** — jogos para calculadoras programáveis" e foi adaptada para o TK para ser rodado em modo "**FAST**" (versão original para calculadora programável FX-702P).

O objetivo consiste em conseguir o maior número de fortunas possível, portanto, ganha aquele que tiver maior número de fortunas em jogo após o preenchimento das casas vazias (+) ou zerar o adversário dentro de um tabuleiro de 8x8.

Para conseguir estas fortunas é necessário prensar as peças do adversário e fazê-lo entregar a você, assim temos as seguintes regras:

a) Só é possível adquirir suas fortunas prensando uma ou mais peças do adversário (**O**) em duas extremidades tanto nas diretas como diagonais, digitando para isso as coordenadas onde você deseja colocar a sua peça, primeiro na horizontal e depois na vertical o computador fará o mesmo após a sua jogada.

Exemplo:

X = **? 4** depois **NEW LINE**
Y = **? 5** depois **NEW LINE**

b) Caso não seja possível realizar o ítem anterior, isto é, fique sem saída, não tendo lugar para colocar sua peça, é necessário passar a sua vez, digite um número negativo.

Exemplo:

X = **? −1** depois **NEW LINE.**

```
 1 REM EDSON MIKIO YOSHIDA
                E
       NELSON MURASSAKI
 2 DIM C$(16,8)
 3 DIM A$(8,8)
 4 LET P=1
 5 GOTO 97
 6 PRINT AT 16,2;"              "
 7 PRINT AT 17,2;"X="
 8 INPUT X
 9 LET Z=X+1
10 GOSUB 21
11 LET X=INT Z
12 PRINT AT 17,2;"Y="
13 INPUT Y
14 PRINT AT 17,2;"   "
15 LET Z=Y+1
16 GOSUB 21
17 LET Y=INT Z
18 IF A$(X,Y)="+" THEN GOTO 28
19 PRINT AT 11,5;"OUTRA VEZ"
20 GOTO 6
21 IF Z<=0 THEN PRINT AT 11,5;
"PASSE     "
22 IF Z<=0 THEN PAUSE 100
23 POKE 16437,255
24 PRINT AT 11,5;"              "
```

continua

```
25 IF Z<1 THEN GOTO 36
26 IF Z<=8 THEN RETURN
27 GOTO 19
28 LET B$="■"
29 LET G$="O"
30 LET P=0
31 LET Q=0
32 PRINT AT 16,2;"VOCE ";X-1;" ";Y-1
33 GOSUB 51
34 PRINT AT 5,2;"          "
35 GOSUB 81
36 LET B$="O"
37 LET G$="■"
38 LET P=1
39 LET Q=0
40 FOR S=1 TO 16
41 FOR R=8 TO 1 STEP -2
42 LET X=VAL C$(S,R)
43 IF X=9 THEN GOTO 6
44 LET T=R-1
45 LET Y=VAL C$(S,T)
46 IF A$(X,Y)="+" THEN GOSUB 51
47 IF Q=1 THEN PRINT AT 5,2;"COMP. ";X-1;" ";Y-1
48 IF Q=1 THEN GOTO 61
49 NEXT R
50 NEXT S
51 FOR I=-1 TO 1
52 FOR J=-1 TO 1
53 LET E=X
54 LET F=Y
55 LET M=X
56 LET N=Y
57 IF I=0 THEN IF J=0 THEN GOTO 59
58 IF E+I>=1 THEN IF E+I<=8 THEN IF F+J>=1 THEN IF F+J<=8 THEN GOTO 63
59 NEXT J
60 NEXT I
61 IF P=1 OR Q=1 THEN RETURN
62 GOTO 19
63 IF A$(E+I,F+J)<>B$ THEN GOTO 59
64 LET E=E+I
65 LET F=F+J
66 IF E>=1 THEN IF E<=8 THEN IF F>=1 THEN IF F<=8 THEN GOTO 68
67 GOTO 59
68 LET D$=A$(E,F)
69 IF D$<>B$ THEN GOTO 71
70 GOTO 64
71 IF D$="+" THEN GOTO 59
72 LET A$(X,Y)=G$
73 LET M=M+I
74 LET N=N+J
75 LET T$=A$(M,N)
76 IF A$(M,N)=B$ THEN LET A$(M,N)=G$
77 IF T$=B$ THEN GOTO 72
78 LET Q=1
79 GOTO 59
80 PRINT AT 3,17;"0 1 2 3 4 5 6 7"
81 LET Y=0
82 LET C=0
83 FOR J=1 TO 8
84 PRINT AT J*2+3,15;J-1;
85 FOR I=1 TO 8
86 PRINT " ";A$(I,J);
87 IF A$(I,J)="O" THEN LET Y=Y+1
88 IF A$(I,J)="■" THEN LET C=C+1
89 NEXT I
90 NEXT J
91 PRINT AT 0,0;"SCORE: VOCE=";Y,"COMP.= ";C;"  "
92 IF Y=0 OR C=0 OR Y+C=64 THEN GOTO 123
93 IF P=0 THEN PAUSE 300
94 POKE 16437,255
95 IF P=0 THEN RETURN
96 GOTO 6
97 FOR J=1 TO 8
98 FOR I=1 TO 8
99 LET A$(I,J)="+"
100 NEXT I
101 NEXT J
102 LET A$(4,4)="■"
103 LET A$(5,5)="■"
104 LET A$(4,5)="O"
105 LET A$(5,4)="O"
106 LET C$(1)="88118118"
107 LET C$(2)="66366333"
108 LET C$(3)="34655643"
109 LET C$(4)="46355364"
110 LET C$(5)="31415161"
111 LET C$(6)="13141516"
112 LET C$(7)="83848586"
113 LET C$(8)="38485868"
114 LET C$(9)="32425262"
115 LET C$(10)="37475767"
116 LET C$(11)="23242526"
117 LET C$(12)="73747576"
118 LET C$(13)="22777227"
119 LET C$(14)="12217182"
120 LET C$(15)="17287887"
121 LET C$(16)="99999999"
122 GOTO 80
123 PRINT AT 21,0;"DESEJA JOGAR NOVAMENTE ? (S/N)"
124 PAUSE 10000
125 POKE 16437,255
126 CLS
127 IF INKEY$="S" THEN GOTO 1
128 STOP
129 SAVE "OTELL■"
130 GOTO 2
```

PROGRAMA DO LEITOR

Dineu Balduino A. Oliveira

Este programa pode ser rodado com apenas 1K de memória. Com a utilização de um símbolo gráfico (graphics 6) disposto em seis posições diferentes, formando símbolos do I CHING.

O I CHING é um livro que ajuda a prever o futuro.

Este livro é difundido em todo o mundo; existem adaptações e traduções em diversas línguas. É considerado uma das obras básicas da literatura chinesa, além de ter inspirado as correntes religiosas do taoísmo e do confucionismo.

Na realidade, o I CHING não pode responder quem vai ganhar na loteria nem se vai chover amanhã. Mas pode dar conselhos e sugerir reflexões, ajudando as pessoas a se decidirem sobre o que fazer e que soluções escolher.

As origens deste livro não são muito claras, mas o texto básico da obra foi elaborado pelo rei Wên e seu filho, o Duque Chou, por volta do ano 1150 a.C.

O livro é formado de 64 signos, seguidos de comentários e sugestões para a sua interpretação. Para quem não está muito familiarizado com o I CHING, as respostas e os conselhos nem sempre são facilmente entendidos. É necessária uma interpretação pessoal de cada resposta encontrada. Segundo uma tradição, essas respostas foram conseguidas através da leitura de símbolos encontrados nos cascos das tartarugas.

Como os comentários a respeito dos símbolos são muito extensos, após o programa (fig. 1) temos uma pequena descrição de cada símbolo.

Finalizando a explicação segue o programa:

```
  15 PRINT "I CHING"
  20 PRINT "QUAL E▀ O SEU NOME ?
"
  25 INPUT A$
  30 GOSUB VAL "85"
  35 PRINT "E O SOBRENOME ?"
  37 INPUT B$
  40 PRINT "SEU HEXAGRAMA DE HOJ
E E▀:"
  45 FOR D=VAL "1" TO VAL "6"
  50 LET C=(CODE A$*CODE B$)*RND
>CODE A$*CODE B$/2
  55 IF C THEN PRINT Y$
  60 IF NOT C THEN PRINT F$
  65 NEXT D
  70 PRINT "SEU NUMERO DE SORTE
E▀ ";INT (CODE A$*CODE B$*RND)+1
  75 PRINT "VEJA O I CHING PARA
SUA INTER-  PRETACAO."
  80 STOP
  85 LET Y$="▬ ▬"
  90 LET F$="▬▬▬"
  95 RETURN
```

fig. 1

O MAPA DA RAM DO TK

Pierluigi Piazzi

Esta matéria originou-se de uma carta enviada pelo assinante Roberto R.Malcher (veja seção DESGRILANDO na pág. 4). Ao invés de respondermos rapidamente para eliminar uma dúvida particular do leitor, resolvemos explorar o assunto de maneira mais completa.

A memória do TK está organizada como uma pilha de bytes, cada um correspondendo a um número cujo valor está entre Ø e 255. Cada posição da memória tem um endereço que vai de Ø até um valor que depende da RAM disponível no computador. Lembrando que 1 Kbyte corresponde a 1Ø24 bytes ($2^{1Ø}$) e que a ROM do TK82 ocupa 8 kbytes, a RAM deveria começar no endereço 8192.

Por certas particularidades do projeto original os endereços de 8192 a 16383 não são acessados pelo sistema operacional, mas estão reservados para futuras expansões da ROM (o que já ocorreu, aliás, no TK85). Isto significa que o primeiro endereço da RAM será 8K+8K=16x1Ø24 = 16384.

Do endereço 16384 ao endereço 165Ø8 estão armazenados os valores das variáveis do sistema operacional do TK. Estes 125 bytes são um pedacinho da RAM que a ROM usa para fazer o computador funcionar. A partir do endereço 165Ø9 fica armazenado o programa. Até que endereço? Ora este valor é ***variável,*** pois o programa que foi inserido na RAM pode ter vários comprimentos. Uma das variáveis armazenadas nos 125 bytes iniciais é justamente a que indica o endereço do byte seguinte àquele em que termina o programa.

Lendo estas variáveis, podemos determinar as fronteiras entre uma região de memória e outra.

O fim da memória, por exemplo, é dado por uma variável batizada RAMTOP contida nos bytes de endereços 16388 e 16389.

Num computador com 2K de RAM o fim da memória está no endereço 18432, ou seja, 16384 + 2K.

A variável **RAMTOP,** então, vale 18432. Como em dada byte cabe, no máximo, um número até 255 usa-se a seguinte convenção: o valor armazenado no segundo byte (no caso 16389) é 256 vezes o número nele contido.

Para armazenar, por exemplo, o número 195Ø6 em 2 bytes, devemos escrevê-lo assim:

1º byte = 5Ø
2º byte = 76

pois,

5Ø + 256 x 76 = 195Ø6

O valor da variável RAMTOP para 2K, ou seja 18432 deve então ser escrito:

1º byte = Ø
2º byte = 72

pois,

Ø + 256 x 72 = 18432

Experimente digitar no seu computador

```
PRINT PEEK 16388,PEEK 16389
```
fig. 1

você deverá obter:

Ø　　　72

para 2K de RAM. Se o seu computador tiver 16K de RAM você obterá:

Ø　　　128

pois,

Ø + 256 x 128 = 32768

continua

Lembrando que a RAM do TK começa no endereço 16384, digite o programa da figura 2.

```
  10 LET RAMTOP=PEEK 16388+256*P
EEK 16389
  20 LET RAMDISP=RAMTOP-16384
  30 PRINT "RAMTOP=";RAMTOP
  40 PRINT "RAM DISPONIVEL=";RAM
DISP/1024;" KBYTES"
```

fig. 2

Se você tiver 2K, obterá:

```
RAMTOP=18432
RAM DISPONIVEL=2 KBYTES
```

fig. 3

Se você tiver 16K, obterá:

```
RAMTOP=32768
RAM DISPONIVEL=16 KBYTES
```

fig. 4

Se o seu computador for um TK85 com 48K (RAM) ou um TK82 com expansão para 64K (ROM + RAM) você deverá obter:

```
RAMTOP=65535
RAM DISPONIVEL=47.999023 KBYTES
```

fig. 5

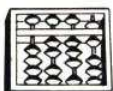
Se você não obtiver este valor, leia a resposta à carta do Roberto na pág. 4 . Vejamos agora como está organizada a RAM, desde o endereço 16384 até a RAMTOP:

INÍCIO DA RAM	RAMTOP CODR DFILE SEMT VARS	VARIÁVEIS DO SISTEMA	16384
		2 BYTES LIVRES	
INÍCIO DO PROGRAMA	10 LET A=5 20 PRINT A 22 PRINT 25 PRINT"A=";A	PROGRAMA	16509
DFILE		ARQUIVO DA TELA DE TV	PEEK 16396 + 256 * PEEK 16397
VARS	A = 0 B$ = "MICROHOBBY" DIM C$ (3,4)	VARIÁVEIS	PEEK 16400 + 256 * PEEK 16401
		1 BYTE CH$ 28	
ELINE		LINHA SENDO DIGITADA + ESPAÇO DE TRABALHO	PEEK 16404 + 256 * PEEK 16405
PILFUN		PILHAS PARA CÁLCULOS	PEEK 16410 + 256 * PEEK 16411
PILFIM		ESPAÇO DE MEMÓRIA	PEEK 16412 + 256 * PEEK 16413
	Z80	PILHA PARA O MICROPROCESSADOR	NÃO ACESSÍVEL EM BASIC
SP	GOSUB RETURN GOSUB RETURN GOSUB RETURN GOSUB RETURN	PILHA PARA ENDEREÇAR SUBROTINAS	PEEK 16386 + 256 * PEEK 16387
RAMTOP	USR	ESPAÇO QUE PODE SER RESERVADO PARA ROTINAS EM LINGUAGEM DE MÁQUINA	PEEK 16388 + 256 * PEEK 16389

fig. 6

Estudando este mapa da RAM, vemos que temos várias maneiras de medir o comprimento de um programa, ou a memória necessária para rodá-lo. Vamos então acrescentar algumas linhas ao programa da figura 2 para ampliá-lo até o indicado na fig. 7:

```
  10 LET RAMTOP=PEEK 16388+256*P
EEK 16389
  20 LET RAMDISP=RAMTOP-16384
  30 PRINT "RAMTOP =";RAMTOP
  40 PRINT "RAM DISPONIVEL =";RA
MDISP/1024;" KBYTES"
  50 LET DFILE=PEEK 16396+256*PE
EK 16397
  60 LET PROGRAMA=DFILE-16509
  70 PRINT "PROGRAMA =";PROGRAMA
;" BYTES"
  80 LET VARS=PEEK 16400+256*PEE
K 16401
  90 LET TELA=VARS-DFILE
 100 PRINT "TELA =";TELA;" BYTES
"
```

fig. 7

Ao rodar este programa você obterá o indicado na fig. 8, se sua máquina tiver 1 ou 2K e o indicado na fig. 9 se tiver mais que 3,5K de RAM.

```
RAMTOP =18432
RAM DISPONIVEL =2 KBYTES
PROGRAMA =366 BYTES
TELA =81 BYTES
```

fig. 8

```
RAMTOP =32768
RAM DISPONIVEL =16 KBYTES
PROGRAMA =366 BYTES
TELA =793 BYTES
```

fig. 9

Isto ocorre porque o sistema do TK tenta economizar o máximo de RAM quando esta é pouca (menor que 3,5K). O truque usado consiste em armazenar no arquivo da tela apenas o que está impresso, não consumindo memória com espaços vazios (fig. 10).

Isto gera um cuidado, ao medir um programa elaborado para um computador com 2K ou 1K, fazer isto com a expansão desconectada.

Finalizando, vamos resumir como medir a memória utilizada:

1) MEMÓRIA OCUPADA SÓ PELO PROGRAMA:

PRINT PEEK 16396 + 256 * PEEK 16397 – 16509

2) MEMÓRIA OCUPADA PELO PROGRAMA, VARIAVEIS, TELA E VARIÁVEIS DO SISTEMA:

PRINT PEEK 16404 + 256 * PEEK 16405 – 16384

3) MEMÓRIA AINDA DISPONÍVEL (incluindo a pilha para o microprocessador):

PRINT (PEEK 16386 + 256 * PEEK 16387) – (PEEK 16412 + 256 * PEEK 16413) + 87

Note que somamos 87 bytes, correspondentes ao comprimento do próprio comando direto.

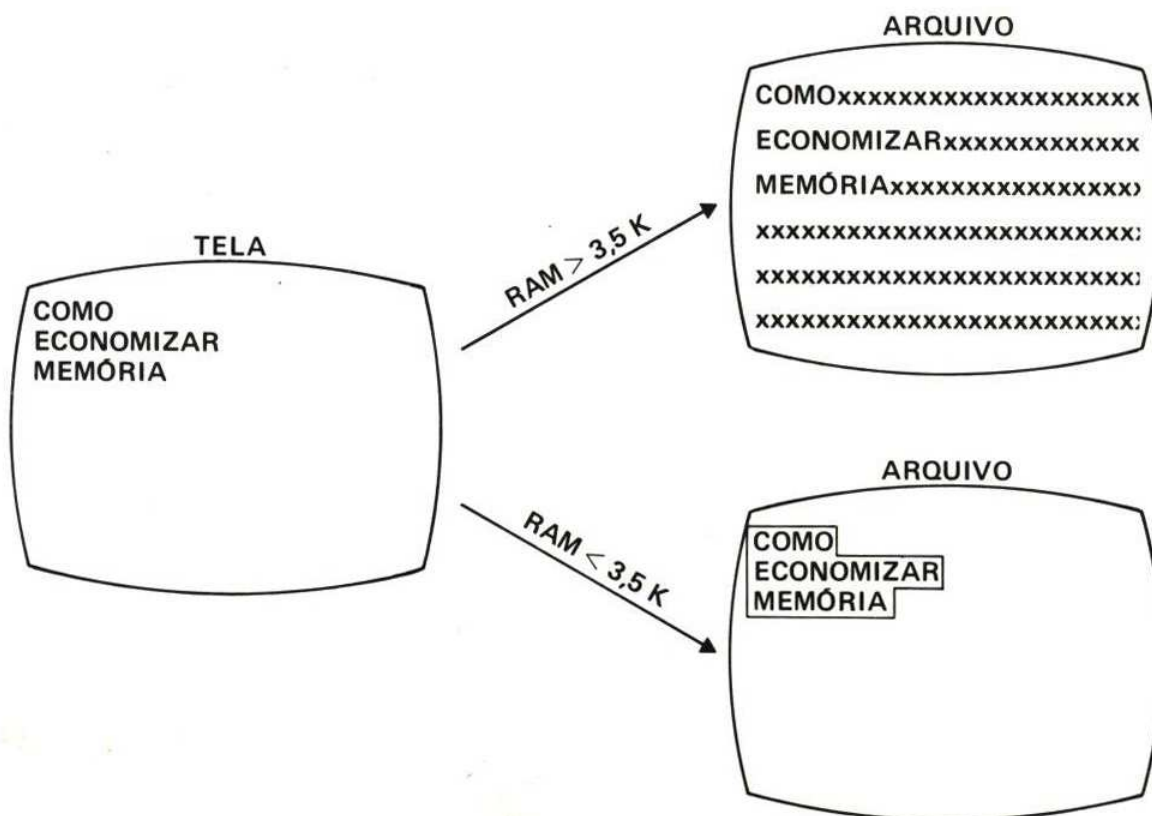



fig. 10

por dentro do apple

® Apple é marca registrada de Apple Computer, Inc.

Prof. Wilson José Tucci – Coordenador de Projetos Especiais da Escola Experimental Pueri Domus.

Apresentando os Apple

A Microhobby é a revista dos usuários do TK, um excelente computador para nos iniciarmos e explorarmos o mundo da computação. Entretanto você pode estar pretendendo comprar um computador mais sofisticado, ou que você, senhor Executivo, tenha em sua firma tal computador. Por isso estamos abrindo uma nova seção, que vai tratar do mais popular, e um dos mais sofisticados, computadores dos Estados Unidos, o Apple, do qual já existem diversos similares no Brasil.

São muitas as diferenças entre o TK e o Apple, vamos aqui, procurar explicar algumas. Ao longo das publicações iremos 'destrinchando' o Apple.

No campo do Hardware são essas as maiores diferenças entre os dois computadores:

	TK	Apple
Memória	2K – 64K	16K – 64K
Processador	Z8Ø, 2.56 MHz	65Ø2, 1.Ø23 MHz
Tela	controlado pelo processador	controlada por circuitos especializados
Gráficos	baixa resolução preto e branco	baixa e alta resolução, coloridos
Caracteres	alfanuméricos e gráficos	alfanuméricos

Embora o máximo de memória RAM do Apple seja 'normalmente' 64K, existem circuitos especiais para expandí-la ainda mais.

Para quem não sabe, o símbolo MHz quer dizer milhões de ciclos por segundo, e indica a velocidade com que 'anda' o processador. Uma instrução tipo 'soma' do 65Ø2 pode ser feita em dois ciclos, isto é, o 65Ø2, processador do Apple pode realizar até 5ØØ,ØØØ somas por segundo. A mais longa instrução do 65Ø2 leva 7 ciclos.

Como você já deve ter percebido, então, o 65Ø2 corre mais devagar do que o Z8Ø, e, além disso, tem menos recursos; mesmo assim, o Apple é mais rápido que o TK, pela seguintes razões: se você já abriu o TK, você já viu que ele tem poucos CI's, portanto o Z8Ø tem que fazer quase tudo, inclusive controlar o video (porque no modo FAST não tem video?), já no Apple existem cerca de 1ØØ circuitos integrados para ajudar o 65Ø2, e portanto o microprocessador só tem que se preocupar quase que exclusivamente em executar o programa; além disso, o BASIC do Apple é mais sofisticado, e roda de uma maneira mais eficiente.

Infelizmente o Apple não traz todos aqueles caracteres gráficos que o TK tem, tanto em alta como em baixa resolução, ele só 'plota' um quadradinho, mas as cores e a alta resolução compensam em muito a falta de caracteres gráficos.

Existem outras importantes diferenças entre o TK e o Apple estão no campo do 'Software', ou programas.

O BASIC do Apple se chama Applesoft, e embora toda a lógica do programa e 90% dos comandos sejam idênticos, existem as seguintes diferenças:

1) Enquanto no TK basta digitar uma tecla para aparecer o comando, no Apple é preciso digitar o comando letra por letra.

2) O comando LET não é obrigatório, ou seja, é válido apenas digitar:

```
10 A = 5
```

3) Podemos colocar mais de um comando numa mesma linha, bastando separá-los com ':'

```
15 A = 5:B = 15:PRINT A + B
```

OBS.: Cuidado se for juntar comandos IF . . . THEN, é melhor não fazê-lo.

4) No Apple o comando NEW apaga toda a memória (de programa e de variáveis), CLEAR apaga as variáveis e HOME apaga a tela. Isto é, HOME corresponde aos CLS no TK.

5) Não existe o comando EDIT no Apple, para modificar uma linha de programa usamos o modo ESC.

6) Embora também existem os comandos PEEK e POKE no Apple, e eles executem a mesma ação que no TK, os efeitos não são os mesmos no Apple, pois como você talvez já saiba, os efeitos dos PEEK's e POKE's dependem muito do Hardware de cada computador e como já foi mostrado, os dois são bem diferentes neste ponto.

7) O Applesoft não verifica linhas de programa quando de sua entrada, portanto os erros de sintaxe só são detectados quando rodamos o programa. Uma linguagem alternativa para o Apple, é que geralmente vem com a unidade de disco é o Integer BASIC, que verifica a sintaxe de cada linha na sua entrada, como o TK, mas, em compensação não trabalha com números reais, apenas inteiros, como o nome já diz.

continua

8) No Applesoft não há necessidade de se dimensionar uma variável 'string' antes de guardar caracteres nela. Por exemplo, enquanto, que no TK faríamos:

```
    1 DIM A$(26)
    2 LET A$="ABCDEFGHIJKLMNOPQRS
TUVWXYZ"
```

No Apple basta fazer:

```
1 A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

Podemos no Apple também fazer matrizes 'string', e nesse caso, cada elemento da matriz pode assumir um 'string' inteiro.

Exemplo:

```
1 DIM A$(1Ø,1Ø)
2 A$(1,2)="MEU NOME"
3 A$(1,3)="E' APPLE"
4 A$(4,5)="SEU NOME"
5 A$(8,9)="E' JOAO"
```

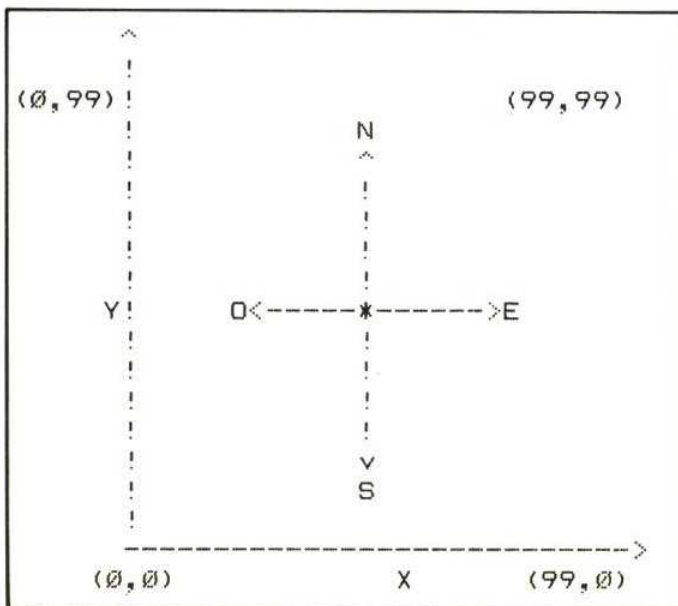
TATUZÃO

Esse pequeno jogo tem como finalidade, além de entreter, introduzir o leitor 'as possibilidades gráficas do APPLE II, e ensinar algumas técnicas para facilitar o desenho e sua movimentação.

Nesse jogo o computador esconde um 'tatu' em algum lugar de um terreno quadriculado de 1ØØ por 1ØØ, e o jogador deve tentar adivinhar a posição do 'tatu' até encontrá-lo.

Após cada tentativa o computador informa ao jogador que direção seguir para encontrar o tatu, quando o tatu é encontrado, ele é desenhado na tela, e aparece o número de chutes que o jogador usou para achar o tatu. O computador então pergunta se quer jogar de novo.

Para se achar o 'tatu' nesse jogo é preciso conhecer duas coisas: os pontos cardeais e colaterais, e sistema de coordenadas x, y.

É assim que o campo quadriculado está organizado nesse jogo:

O centro do quadriculado é o 5Ø,5Ø. Você deve ir chutando as coordenadas de acordo com as indicações do computador. Só na primeira vez é que não há indicação nenhuma. Lembre-se que devemos sempre entrar o X antes do Y (é só lembrar da velha aula da geometria).

Obviamente esse programa não oferece a animação de um 'defender' ou 'commander', mas em compensação pode ser facilmente entendido (acompanhe os REMARKS).

A princípio da técnica para se poder desenhar e mover facilmente uma figura é a seguinte:

Guardamos os dados necessários para o desenho da figura numa matriz (linha 1ØØ). Todos esses dados são relativos a uma posição, no caso o centro da figura. Após determinarmos o ponto onde queremos que fique o centro da figura (variáveis XP e YP) executamos a subrotina 54Ø (ou 58Ø, se for para desenhar o 'tatu'). Essa subrotina lê os dados da matriz respectiva, e desenha. Na verdade cada dado para desenho consiste de quatro números, pois como você já deve ter notado essa matriz executa um HPLOT de um ponto a outro. A técnica de executar HPLOT's de um ponto a outro é bem mais rápida e exige bem menos dados nos comandos DATA do gravar cada ponto do desenho na matriz. Para se apagar uma figura, simplesmente a redesenhamos na mesma posição, só que em preto.

Essa técnica é bem geral e se aplica a muitos computadores, só que no caso de não ser possível o recurso de desenhar uma linha de um ponto a outro, deve-se guardar todos os pontos do desenho (relativos a algum referencial), e trocar a linha 54Ø para:

```
54Ø  FOR I = 1 TO TH STEP 2:
     HPLOT XP + PH%(I + 1),YP + PH%(I):
     NEXT I:
     RETURN
```

O uso de matrizes inteiras se justifica por essas usarem bem menos memória do que usariam matrizes de ponto flutuante.

Nas próximas publicações veremos métodos especiais do APPLE para desenhar e fazer animações de figuras, tão rápidos como os jogos de fliperama que você conhece. Aguarde. . .

continua

```
Ø REM      *** TATUZAO ***

      @1983 POR DENTRO DO APPLE

1Ø  REM DIMENSIONAR AS MATRIZES
2Ø  REM PT% --> PONTOS DO TATU
3Ø  REM PH% --> PONTOS DO HOMEM
4Ø  REM  A$ --> DIRECOES
5Ø  DIM PT%(2ØØ),PH%(2ØØ)
6Ø  REM LER MATRIZES DE PONTOS
7Ø  REM DO HOMEM E DO TATU
8Ø  REM TT --> TAMANHO DO TATU
9Ø  REM TH --> TAMANHO DO HOMEM
1ØØ  READ TT,TH: FOR I = 1 TO TT:
      READ PT%(I): NEXT I: FOR I =
      1 TO TH: READ PH%(I): NEXT I
      : FOR I = 1 TO 8: READ A$(I)
      : NEXT I
11Ø  REM GERAR AS COORDENADAS X,Y

12Ø  REM DA POSICAO DO TATU
13Ø  REM E INICIALIZAR CONTADOR
14Ø  TEXT : HOME : HGR :XT =  INT
      ( RND (1) * 1ØØ):YT =  INT (
      RND (1) * 1ØØ):K = Ø
15Ø  REM POSICIONAR CURSOR E
16Ø  REM GERAR XP,YP INICIAIS
17Ø  REM PARA EVITAR ERRO
18Ø  VTAB 22:XP = 5Ø:YP = 5Ø
19Ø  REM ENTRAR NOVA POSICAO
2ØØ  REM DO HOMEM,APAGAR A VELHA
21Ø  REM DESENHAR A NOVA ,E
22Ø  REM INCREMENTAR CONTADOR
23Ø  INPUT "ENTRE X E Y (DE Ø A 9
      9) ";X,Y:K = K + 1:Y = 99 -
      Y: HCOLOR= Ø: GOSUB 54Ø:XP =
      X * 2 + 4Ø:YP = Y + 3Ø: HCOLOR=
      3: GOSUB 54Ø: IF  NOT ((X =
      XT) AND (Y = YT)) GOTO 34Ø
24Ø  HOME : VTAB 22: PRINT "PARAB
      ENS VOCE ACHOU O TATU ": PRINT
      "EM ";K;" VEZES": PRINT
25Ø  REM DETERMINAR OS PONTOS
26Ø  REM PARA DESENHO RELATIVO
27Ø  REM  DO TATU
28Ø  REM  E DESENHAR O TATU
29Ø XP = XP + 15:YP = YP + 15: GOSUB
      58Ø: REM   DESENHA O TATU
3ØØ  GOTO 45Ø
31Ø  REM INICIO DE COMPARACOES
32Ø  REM PARA DETERMINAR QUE
33Ø  REM DIRECAO ESTA O TATU
34Ø  IF (X = XT) AND (Y < YT) THEN
      J = 4: GOTO 44Ø
35Ø  IF (X = XT) AND (Y > YT) THEN
      J = 1: GOTO 44Ø
36Ø  IF (X < XT) AND (Y = YT) THEN
      J = 7: GOTO 44Ø
37Ø  IF (X > XT) AND (Y = YT) THEN
      J = 8: GOTO 44Ø
38Ø  IF (X > XT) AND (Y > YT) THEN
      J = 3: GOTO 44Ø
39Ø  IF (X > XT) AND (Y < YT) THEN
      J = 6: GOTO 44Ø
4ØØ  IF (X < XT) AND (Y < YT) THEN
      J = 5: GOTO 44Ø
41Ø  IF (X < XT) AND (Y > YT) THEN
      J = 2: GOTO 44Ø
42Ø  REM IMPRESSAO DAS
43Ø  REM ORINTACOES AO JOGADOR
44Ø  HOME : VTAB 21: PRINT "VOCE
      ESTA EM ";X;",";99 - Y: PRINT
      "VA PARA ";A$(J): FOR I = 1 TO
      5ØØ: NEXT : GOTO 23Ø
45Ø  PRINT "QUER JOGAR DE NOVO ?
      (S/N) ";: INPUT "";R$: IF  LEFT$
      (R$,1) = "S" THEN 14Ø
46Ø  TEXT : HOME : VTAB 5: PRINT
       TAB( 5)"FOI UM PRAZER": VTAB
      1Ø: PRINT  TAB( 15)"JOGAR CO
      M VOCE": VTAB 15: PRINT  TAB(
      25)"APPLE II"
47Ø  COLOR= 1Ø: HLIN Ø,39 AT Ø: VLIN
      Ø,47 AT 39: HLIN Ø,39 AT 46:
       VLIN Ø,47 AT Ø
48Ø  REM ESPERAR POR UMA TECLA
49Ø  IF  PEEK ( - 16384) < 128 THEN
      49Ø
5ØØ  POKE  - 16368,Ø: HOME : END
51Ø  REM SUBROTINA QUE DESENHA
52Ø  REM O HOMEM EM RELACAO
53Ø  REM AO PONTO XP,YP
54Ø  FOR I = 1 TO TH STEP 4: HPLOT
      XP + PH%(I + 1),YP + PH%(I) TO
      XP + PH%(I + 3),YP + PH%(I +
      2): NEXT I: RETURN
55Ø  REM SUBROTINA PARA DESENHAR
56Ø  REM O TATU EM RELACAO
57Ø  REM  AO PONTO XP,YP
58Ø  FOR I = 1 TO TT STEP 4: HPLOT
      XP + PT%(I + 1),YP + PT%(I) TO
      XP + PT%(I + 3),YP + PT%(I +
      2): NEXT I: RETURN
59Ø  DATA   48,132: REM   NUMEROS
       DE DADOS PARA DESENHAR TATU
       E HOMEM
6ØØ  REM BLOCO DE DADOS PARA
61Ø  REM DESENHO DO TATU
62Ø  DATA  -4,-2,-4,2,-3,-3,-3,3,
      -2,-4,-2,4,-1,-8,-1,5,-3,-6,
      -2,-6,Ø,-1Ø,Ø,6,1,-11,1,7,2,
      -6,2,8,3,-4,3,-3,3,3,3,4,4,-
      5,4,-4,4,2,4,3
63Ø  REM BLOCO DE DADOS PARA
64Ø  REM DESENHO DO HOMEM
65Ø  DATA     -15,-3,-15,2,-14,-4,
      -14,3,-13,-5,-13,4,-12,-4,-1
      2,3,-11,-3,-11,2,-1Ø,-3,-1Ø,
      2,-9,-1,-9,Ø,-8,-1,-8,Ø,-7,-
      4,-7,3,-6,-6,-6,5
66Ø  DATA  -5,-7,3,-7,-5,-6,3,-6,
      -5,5,8,18,-5,6,1,12,-5,-4,-5
      ,3,-4,-4,-4,3,-3,-4,-3,3,-2,
      -4,-2,3,-1,-4,-1,3,Ø,-4,Ø,3,
      1,-4,1,3,2,-4,2,3,3,-4,3,3,4
      ,-4,4,3
67Ø  DATA  5,-4,12,-4,5,-3,12,-3,
      5,-2,12,-2,5,1,12,1,5,2,12,2
      ,5,3,12,3,1Ø,15,8,17,9,17,7,
      19,6,19,5,2Ø
68Ø  REM DIRECOES A SEGUIR
69Ø  DATA  NORTE,NORDESTE,NOROEST
      E,SUL,SUDESTE,SUDOESTE,LESTE
      ,OESTE
```

CALENDÁRIO PERPÉTUO

Quantas vezes acontece de precisarmos de um calendário antigo, de alguns anos ou mesmo meses atrás e, por um motivo qualquer, não encontrarmos nenhum por perto? Quem nunca perguntou a si mesmo algo como "Em que dia da semana vai cair o Natal no ano que vem?" O programa que apresentamos este mês pretende auxiliar os leitores que já tiveram uma dúvida como essa, permitindo-lhes ver, na tela do seu computador, o calendário completo de qualquer mês e ano, e também servir de base para quem quiser aplicar algumas das técnicas básicas usadas no programa.

continua

```
Ø REM ** CALENDARIO PERPETUO **

 @1983 POR DENTRO DO APPLE

1Ø HOME : VTAB 5: HTAB 1Ø: PRINT
    "CALENDARIO PERPETUO"
2Ø INVERSE : VTAB 23: PRINT TAB(
    7);"DIGITE 'Ø,Ø' PARA TERMIN
    AR"; TAB( 41);: NORMAL
3Ø VTAB 12: HTAB 1: PRINT "DIGIT
    E O MES E O ANO, SEPARADOS P
    OR UMA": VTAB 14: INPUT "VIR
    GULA (EXEMPLO: 6,1965): "
    ;MES,ANO: IF MES = Ø AND ANO
     = Ø THEN HOME : END
4Ø RESTORE : FOR I = 1 TO MES: READ
    MES$,NDIAS: NEXT I:MES$ = ME
    S$ + " " + STR$ (ANO)
5Ø REM ANO BISSEXTO?
6Ø IF MES = 2 AND INT (ANO / 4)
     = ANO / 4 THEN NDIAS = 29
7Ø REM CALCULAR EM QUE DIA CAI
          O PRIMEIRO DO MES.
          Ø=DOMINGØ, 1=SEGUNDA...
8Ø IF MES < 3 THEN MES = MES + 1
    2:ANO = ANO - 1
9Ø N = 1 + 2 * MES + INT (.6 * (
    MES + 1)) + ANO + INT (ANO /
    4) - INT (ANO / 1ØØ) + INT
    (ANO / 4ØØ) + 2:N = INT ((N
    / 7 - INT (N / 7)) * 7 + .
    5): IF N = Ø THEN N = 7
1ØØ N = N - 1
11Ø REM MOSTRAR CALENDARIO
12Ø HOME : COLOR= 1Ø: HLIN Ø,39 AT
    Ø: VLIN Ø,47 AT 39: HLIN Ø,3
    9 AT 46: VLIN Ø,47 AT Ø
13Ø VTAB 4: HTAB 2Ø - LEN (MES$
    ) / 2: PRINT MES$: VTAB 8: HTAB
    8: PRINT "D   S   T   Q   Q
     S   S": PRINT
14Ø DIA = 1
15Ø HTAB 4 * N + 8: PRINT DIA;
16Ø DIA = DIA + 1: IF DIA > NDIAS
     THEN 2ØØ
17Ø N = N + 1: IF N > 6 THEN PRINT
    : PRINT :N = Ø
18Ø GOTO 15Ø
19Ø REM ESPERAR POR UMA TECLA
2ØØ IF PEEK ( - 16384) < 128 THEN
    2ØØ
21Ø POKE - 16368,Ø: GOTO 1Ø
22Ø REM DADOS DO PROGRAMA
23Ø DATA JANEIRO,31,FEVEREIRO,2
    8,MARCO,31,ABRIL,3Ø,MAIO,31,
    JUNHO,3Ø,JULHO,31,AGOSTO,31,
    SETEMBRO,3Ø,OUTUBRO,31,NOVEM
    BRO,3Ø,DEZEMBRO,31
```

O programa começa pedindo o mês e o ano e procurando, através do "loop" da linha 4Ø, o nome do mês e o número de dias neste. No caso de se tratar do mês de fevereiro de um ano bissexto (divisível por quatro), o número de dias é corrigido para 29. A rotina definida pelas linhas 8Ø a 1ØØ fazem o cálculo do dia da semana em que começa o mês em questão.

A linha 12Ø mostra uma maneira interessante (e rápida) para colocar na tela caracteres repetidos. Podemos obter efeitos diferentes usando outras cores, "plotando" em linhas pares ou ímpares ou na vertical, ou usando combinações de linhas adjacentes. Mais adiante, o computador entra num ciclo para colocar na tela os números nos seus lugares corretos. Terminando o programa, temos uma sequencia de duas linhas (2ØØ e 21Ø) que prendem o computador num loop até que o operador pressione qualquer tecla.

Finalmente, uma dica para quem quiser usar a fórmula da linha 9Ø em seus programas, para calcular o dia da semana de um dia qualquer: substitua o "1" no início da fórmula pela variável "DIA" (N = DIA + 2 * MÊS. . .). Não se esqueça de incluir a linha 8Ø, que efetua um ajuste nas variáveis MÊS e ANO para os meses de janeiro e fevereiro.

TWO-LINERS

Um dos nossos amigos mais próximos é o Tio Fred List. É um velhinho simpático e um apaixonado hobista em computação. Entretanto, o Apple do nosso querido tio apresenta somente Ø.6K bytes de memória de programação e necessita de alguns programas para ele. Eis aqui um desafio. . . Vamos ajudá-lo.

No BASIC usado no Apple (Applesoft ou Integer) é permitida uma instrução de múltiplos comandos, desde que eles venham separados por dois pontos; e assim, as instruções:

```
1Ø HOME
2Ø PRINT "ISTO E' UM TESTE"
3Ø PRINT
4Ø FOR I = 1 TO 1Ø
5Ø PRINT I;" ";
6Ø NEXT I
7Ø PRINT "FIM"
8Ø END
```

continua

podem ser escritas em uma única linha:

```
10  HOME : PRINT "ISTO E' UM TESTE":
    PRINT : FOR I = 1 TO 10: PRINT I;
    " ";: NEXT I: PRINT "FIM": END
```

Entretanto, uma linha só aceita até 255 caracteres. . . cuidado!

Para o desafio dos TWO-LINERS vamos estabelecer algumas regras:

todos os programas devem ser escritos em Applesoft ou Integer BASIC e não devem ter mais que duas linhas de programação;

não há limites para o número de múltiplos comandos ou comprimento de cada linha, desde que, evidentemente, a linha possa ser digitada diretamente pelo teclado.

Favor enviar, para a apreciação do nosso staff, uma cópia legível do seu TWO-LINER. Essa cópia, que não será devolvida, poderá ser usada pelo departamento como e quando o julgarmos necessário.

Os programas serão julgados com os seguintes critérios:

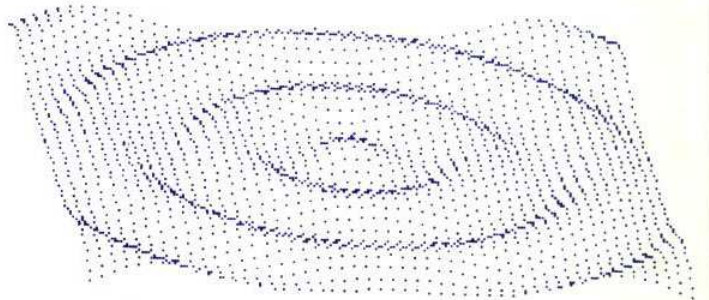
a) impacto que o programa apresenta ao rodá-lo.

As decisões do staff do POR DENTRO DO APPLE são semi-finais. Tio Fred List tem a última palavra.

PRÊMIOS:

1º colocado: um livro sobre apples.
2º colocado: uma foto do Corinthians
3º ao 9º colocados: ainda não decidimos
10º colocado: uma foto do Tio Fred List

Eis aqui dois exemplos feitos por Mark Bachman e Christopher Volpe:


```
1  HGR : FOR Z = 0 TO 10 STEP .2:
     FOR X = 0 TO 10 STEP .2: HCOLOR=
     3:Y =  - 10 * COS (3 * SQR
     ((X - 5) ^ 2 + (Z - 5) ^ 2))
     / 2 + 50: HPLOT X * 20 + 20
     + Z * 3,Y + Z * 10
2  HCOLOR= 0: HPLOT X * 20 + 20 +
     Z * 3,Y + Z * 10 + 1 TO X *
     20 + 20 + Z * 3,180: NEXT : NEXT
     : END
```

```
1 H$ = "303:AD 52 C0 AD 51 C0 A9
     22 EA EA EA EA 20 A8 FC AD 5
     0 C0 A9 47 3E 00 00 3E 00 00
     3E 00 00 3E 00 00 3E 00 00
     3E 00 00 3E 00 00 3E 00 00 3
     E 00 00 3E 00 00 20 A8 FC 4C
     06 03 N 3036"
2  FOR I = 1 TO 30: PRINT "PAGINA
     UM DE TEXTO  ";: NEXT : FOR
     C = 1 TO  LEN (H$): POKE 511
     + C, ASC ( MID$ (H$,C,1)) +
     128: NEXT : CALL  - 144
```

Já vimos os exemplos, portanto, mãos à obra e enviem os seus programas para:

MICROHOBBY
Seção: Por Dentro do Apple
CX. POSTAL 60081

STAFF:

José Eduardo Moreira
Daniel R. Falconer

FUNÇÕES I

A fita FUNÇÕES I é comercializada pela MICRON de São José dos Campos (av. São João, 74) e tem como principal finalidade a de ampliar o sistema operacional do TK com algumas funções adicionais extremamente úteis. Vem acompanhada de um manual que, se peca um pouco pelo aspecto gráfico, compensa isso amplamente pela boa didática empregada nas instruções.

Ao ser carregado normalmente o programa já sai rodando apresentando o menu da fig. 1.

```
MICRON ELETRONICA
  FUNCOES EXTRAS

MEMORIA USADA S/ VARIAVES E VD.
PRINT USR 32440

APAGADOR DE PROGRAMA
INCLUA NO PROGRAMA;
LET L=USR 32465
E ELE SERA APAGADO DESTA
LINHA ATEH O FINAL

RENUMERADOR DE LINHAS
INCLUA NO PROGRAMA;
1 REM
2 LET L=USR 32000
3 LIST
IMPORTANTE: LINHA 1 REM CONTEM:
2 ESPACOS, 1 GRAFICO, 1 ESPACO.
USE POKE 16516,1 OU 5 OU 10.
    PROIBIDO REPRODUZIR
      DIGITE NEW
```

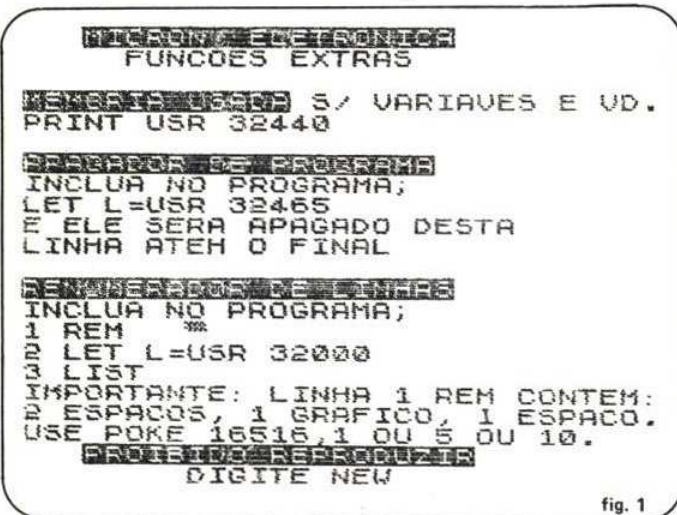



fig. 1

Após ler as instruções, o usuário deve digitar **NEW:** o programa se auto-carrega no último Kbyte dos 16K de RAM.

Para testar a fita carregamos no computador o programa MINI-INVASOR com o programa RENUMERANDO "enganchado" no fim (publicados, ambos, no número 1 da MICROHOBBY).

A primeira tarefa da fita FUNÇÕES foi a de eliminar o programa RENUMERANDO do fim do MINI-INVASOR.

Como o programa MINI-INVASOR termina na linha 1135 e o RENUMERANDO começa na linha 9959, basta então acrescentar a linha

```
1136 LET L=USR 32465
```

e a seguir

```
GOTO 1136
```

Em poucos segundos o RENUMERANDO em BASIC desaparece, deixando apenas o programa MINI-INVASOR.

A seguir testamos o RENUMERANDO da própria fita (em ASSEMBLY). Para isso devemos acrescentar

```
1 REM
2 LET L=USR 32000
3 LIST
```

A linha 1 REM deve contar 4 espaços em branco.

A seguir devemos comandar

```
POKE 16516,N
```

onde N é o incremento que queremos dar aos endereços da nova renumeração até 20 no máximo). Como o MINI-INVASOR estava renumerado de 5 em 5, alteramos para que o fosse de 20 em 20.

```
POKE 16516,20
```

Rodamos o programinha (com RUN), obtemos a renumeração desejada num tempo incrivelmente curto! Isso ocorre porque, ao contrário do programa RENUMERANDO em BASIC que publicamos no número 1, este é em linguagem de máquina. Finalmente, para saber quantos bytes foram consumidos pelo programa basta digitar

```
PRINT USR 32440
```

Esta rotina pode ser alterada fazendo-se

```
POKE 32444,A
```

onde A deve assumir os seguintes valores:

A = 2Ø: consumo de programa, video e variáveis

A = 28: idem acima mais pilha de calculador

A = 12: só o programa (volta ao original)

Em resumo, esta fita contém uma típica rotina "ferramenta" extremamente útil para programadores. Sua utilidade ficou evidenciada quando percebemos que ela entrou no dia-a-dia dos nossos analistas de SOFTWARE aqui na redação.

NOVIDADES

Nós da MICROMEGA PUBLICAÇÕES E MATERIAL DIDÁTICO LTDA., editora da revista MICROHOBBY convidamos todos nossos leitores e assinantes a visitar nosso espaço na III Feira da Informática a se realizar de 17 a 23 de outubro, no Parque Anhembi, em São Paulo.

Na ocasião será distribuida uma tabela-brinde especial aos novos assinantes. Os assinantes antigos também poderão retirar este brinde, entregando, na feira, a etiqueta de endereçamento colada no envelope deste número 4.

Além disso, no nosso stand vamos apresentar uma série de novidades para o TK: a nossa finalidade é a de incentivar pequenas empresas a produzir periféricos e acessórios para o TK, ampliando cada vez mais sua utilidade e desempenho. Vamos ter também sorteios interessantíssimos, demonstrações de novos equipamentos e lançamento de livros para o TK. O

A Memphis, empresa paulistana, líder no mercado de suprimentos para computadores, vem se projetando e se firmando há 14 anos em todo o Território Nacional.

Com a finalidade de manter o seu maior lema "Produtos de Qualidade a Preços Justos", e para melhor atender a sua exigente clientela e a todos os seus amigos e habituais fornecedores, acabou de mudar sua Matriz para um moderno prédio situado à Av. Angélica, 35 no bairro de Santa Cecília, com uma área de 4.000 m², contando com amplo estacionamento.

Nesse local estarão funcionando os Departamentos Administrativo, Financeiro, Estoques, setor Fabril e também o C.P.D. (que a partir de agora agilizará ainda mais o atendimento aos clientes).

O Departamento de Vendas continuará atendendo aos seus clientes em seus escritórios à Av. Arnolfo de Azevedo, 108 — Pacaembú pelo telefone 262-5577 em São Paulo e no Rio de Janeiro à Praia do Flamengo, 66 grupo 1.519 pelos telefones 205-3849 e 225-3469.

"Novo ABC dos Computadores" de Allan Lytel e Carlos Alberto M. Marques, é o mais recente lançamento da editora "Antenna" (C.P. 1131 — Rio de Janeiro — CEP: 20001). É um livro especialmente escrito para as pessoas que desejem ingressar no mundo da informática, seja com finalidades profissionais, seja para quem possua ou pretenda possuir um microcomputador, prestando-se igualmente, aos autodidatas, e, como livro auxiliar, nos cursos de iniciação ao Processamento de Dados.

Compõe-se de 12 capítulos: Computadores Digitais e Analógicos — Números para computadores — Operações Aritméticas — Lógica Simbólica — Circuitos Lógicos Básicos — Multivibradores — Contadores — Circuitos Calculadores — Circuitos Integrados — Armazenamento de Dados — Dispositivos de Entrada e Saída — Programação. É redigido em linguagem acessível aos iniciantes, constando de 176 páginas, com bastante ilustrações, no formato 13,5x21 cm. É vendido nas boas livrarias ao preço de Cr$ 3.500,00 o exemplar. O

A Computronix, uma loja de microcomputadores de Belo Horizonte está lançando o teclado profissional para o TK82-C e o NEZ 8000. Trata-se de caixa em madeira ou fibra de vidro que comporta o aparelho, a expansão e a fonte internamente, operando-se em teclado tipo IBM e um vídeo 12" por cima da caixa. Maiores informações: fone: (031) 225-3305. O

A MICRODIGITAL lança o TK83, versão sofisticada do TK82-C. Seu desempenho como computador é idêntico, tendo as mesmas características operacionais de seu antecessor. Do ponto de vista estético, porém, ele apresenta um "design" muito mais moderno e funcional. A nova caixa é mais resistente, tornando os circuitos internos menos vulneráveis a pressões mecânicas. O "hardware" foi redesenhado implicando num projeto mais elegante e compacto.

Com esse lançamento, a MICRODIGITAL mostra toda a confiança que deposita na antiga máquina, dissipando de vez os temores de "obsolência" apresentados por alguns usuários quando do lançamento do TK85.

O TK82, agora 83, veio para ficar durante muito tempo, ainda: a relação desempenho/custo é ainda (e parece que será por muito tempo) a mais elevada do mercado nacional.

Num mercado com muitos "cadillacs" enfeitados, um "fusquinha" terá sempre uma grande aceitação, especialmente numa época de economia em contenção. O

PEQUENOS

ANÚNCIOS

Vendo esquema completo do microcomputador NEZ8000 (com SLOW e expansão de memória) – 3 cópias heliográficas. Cheque nominal de Cr$ 6.000,00 para *JAN MARTIN LUND, Rua Frederico Ozanã, 16/21 – 11100 – Santos – SP.*

Troco ou vendo programas (nacionais e importados) para os micros: CP-200, TK82-C e ZX 81. Possuo o Simulador de Vôo, Turbo, Galaxy e muitos outros jogos e aplicativos. Tratar com *Eduardo Medeiros – R. Eliseu Guilherme, 1076 – Ribeirão Preto – SP – CEP: 14100* – Atendo também por reembolso.

Vendo ou troco programas de jogos para TK82-C e TK85. Tratar com *Edson, R. Guariba, 54 – São Paulo – Tel.: 918.0998 (c/ Ruth).*

RADIO MICRO: O primeiro grupo de rádio amadores digitais do Brasil convida todos os radioamadores a conhecer os projetos de hard e soft que estamos desenvolvendo, aos interessados entrar em contato com *PY2-EMI Renato Strauss – R. Cardoso de Almeida, 654/32 – 05013 – São Paulo – SP.*

Colocamos vídeo-reverso em televisores TX-Philips, fazemos adaptações de TV's para monitor de vídeo e vendemos kits de vídeo-reverso. *Luiz Wellington – Tel.: 224-2776 – Fortaleza/CE.*

Vendo programas nacionais e importados, 2K e 16K, BASIC, e ASSEMBLER para micros TK82-C e similares (Aceito trocas) – *Carlos A. Sciaretti – Cx. Postal: 5567 – São Paulo/SP – CEP: 01051 – tel.: 522-8586.*

Curso de Linguagem de máquina para TK82 e TK85 (ASSEMBLY Z80). Aos sábados das 9:00 às 13:00 hs. – São Paulo – *Fone: 813-4555 (Márcia) à tarde.*

Serviço de Datilografia – IBM. Teses, apostilas e manuais técnicos. *Telefone: 258-8486 – Américo – SP.*

Conversor do NEZ8000 para SLOW: Em São Paulo, *WILSON DE ASSIS* faz a inclusão desta função no NEZ8000 por Cr$ 20.000,00 (mais as despesas de embalagem e correio) e dá a garantia de 6 meses. Se o seu endereço for próximo ao dele, ele entregará em casa. O seu *telefone é: (011) 203-7967 e o endereço: Rua Fabricio Correia, 145, Tucuruvi, 02311 – São Paulo – SP.*

Sou possuidor de muitos programas para o TK82. Os interessados em trocá-los escrevam para *Marco Aurélio Dias de Oliveira – Av. Afonso Pena, 1557 – Bloco B – apto. 214 – Campo Grande/MS – CEP: 79100.*

Troca de programas para TK – *Carlos Horácio C. Fontenelle – R. Alfeu Aboim, 770 – Aldeota – Fortaleza – CE – CEP: 60000*

ESPIÃO – Jogo de Aventura para TK e CP-200. Um desafio a sua Memória e Habilidade. 16K com SLOW. Cr$ 5.000,00. Pedidos pelo reembolso para *Cx. Postal: 28 – 27200 – Piraí – RJ.*

Vendo NEZ 8000 com expansão de 16K – *Tenente Dilson – Tel.: (061) 224-1839 – Brasília – DF.*

COMO COLABORAR COM MICROHOBBY

A revista MICROHOBBY foi criada para servir de intercâmbio entre os leitores que participam do mágico mundo da computação.

A característica realmente inovadora do computador pessoal, está em transformar cada consumidor num criador. Aproveite sua criatividade e envie suas colaborações, recebendo remuneração a título de DIREITO AUTORAL.

Se o material enviado for aprovado para publicação, a remuneração será de Cr$ 10.000,00 por programa. Se o programa for muito bom e muito extenso esta quantia será aumentada, a critério da redação, até Cr$ 30.000,00. Esta remuneração será oferecida também para artigos interessantes sobre o mundo do TK.

A maneira ideal de nos enviar o material a ser publicado obedece às seguintes normas:

1) **Nunca** esquecer de colocar nome completo, endereço e número de sua assinatura em **todo** material enviado (fitas, listagens de impressora, envelope, carta, etc).
2) Enviar a listagem de programa **datilografada** ou, melhor ainda, tirada na impressora do TK.
3) Colocar sempre uma linha **REM** com o nome do autor e o título do programa.
4) Enviar uma fita com o programa gravado **algumas vezes** (se possível em gravadores diferentes).
5) Na fita, gravar **com microfone** (em viva voz), algumas instruções úteis, **nome completo e endereço do remetente.**

O material não utilizado não será devolvido. Esta é uma norma corrente em quase todas as revistas do mundo, que gera porém, uma certa desconfiança: "e se apagarem meu nome do programa e o publicarem sem me remunerar?"

Neste ponto podemos sugerir o seguinte:

Ao enviar o material para **qualquer** revista, você, ou confia em sua seriedade, ou, é melhor não enviar o material, reservando-o para uma publicação dígna de seu crédito.

Uma outra saída é a de tirar uma cópia de todo material enviado e registrá-lo em cartório, para qualquer reclamação posterior. Obviamente tal procedimento só é aconselhável se o material enviado for valioso a ponto de justificar toda essa mão-de-obra.

6) O material deve ser enviado para:

MICROMEGA P.M.D. Ltda.
PROGRAMAS DO LEITOR
Cx. POSTAL: 60081
CEP: 05096 – S. PAULO – SP

7) Qualquer dúvida, poderá ser esclarecida pelo telefone:

(011) 257-5767 com Roberto.

Aguardamos sua colaboração.

GERAÇÃO HUMANA NO TK

Antes que o leitor se assuste com visões tipo "Admirável Mundo Novo" pedimos a leitura da parte final da carta que o Roberto R. Malcher de Óbidos (Pará) enviou à Seção DESGRILANDO.

A proposta, pelo que pudemos entender, consiste basicamente em se tentar estabelecer uma correlação entre data de nascimento da mãe, dos filhos e o sexo destes.

Para quem achar absurdo, queremos lembrar que há alguns anos foi realizada na França uma pesquisa que tentava correlacionar o Q.I. de crianças com outros fatores. Verificou-se uma correlação estatisticamente significativa entre o Q.I. e a época do nascimento: crianças nascidas no começo do inverno tinham um desenvolvimento intelectual significativamente superior. Foram levantadas muitas hipóteses (chegou-se até o absurdo de se falar em horóscopo!). A explicação na realidade é bem simples: no inverno europeu os pais ficam mais tempo em casa e essa maior convivência é tremendamente importante para o futuro desenvolvimento mental do bebê.

Por isso pedimos a todos os leitores que tirem uma xerox da tabela anexa (para não recortar a revista) e a enviem preenchida para:

C.R.M.
CP 43
68250 Óbidos PA

Pesquisa Geração Humana

Nasc.: Dia__Mês__Ano __________

Mãe __________ Sexo __________

1º Filho __________

2º Filho __________

3º Filho __________

4º Filho __________

5º Filho __________

6º Filho __________

Obs.: Incluir eventuais filhos falecidos ou abortos.

O FANTASMA

Você está no tenebroso salão de um castelo mal-assombrado. Você conseguiu achar um tesouro e as moedas estão espalhadas no chão. Antes que o fantasma do castelo gele sua alma pecadora, você deve catar a maior quantidade possível de moedas (as da moldura não podem ser recolhidas).

Após digitar RUN você deve escolher o grau de dificuldade, entre .Ø1 (difícil) e .99 (fácil).

O programa roda usando menos de 1K de RAM em SLOW.

O fantasma é um pouco gagá e, enquanto o persegue, come moedas fazendo uma lamentável confusão entre plano espiritual e plano material.

Para se movimentar no salão (tenebroso) do castelo você deve usar as teclas 5, 6, 7 e 8 ou o módulo de comando (Joystick).

Ao ter sua alma congelada pelo fantasma, aparecerá a quantia coletada.

```
  10 LET S=0
  20 CLS
  30 INPUT A
  40 LET Q=1
  50 LET W=1
  60 LET R=INT (RND*7)+1
  70 LET E=INT (RND*7)+1
  80 FOR F=1 TO 10
  90 PRINT "██████████"
 100 NEXT F
 110 IF INKEY$="P" THEN GOTO 20
 120 PRINT AT Q,W;"▒"
 130 IF PEEK (PEEK 16398+256*PEE
K 16399)=155 THEN GOSUB 230
 140 LET Q=Q+(INKEY$="6" AND Q<8
)-(INKEY$="7" AND Q>1)
 150 LET W=W+(INKEY$="8" AND W<8
)-(INKEY$="5" AND W>1)
 160 IF Q=R AND W=E THEN GOTO 26
0
 170 IF RND<A THEN GOTO 110
 180 PRINT AT R,E;CHR$ 11
 190 IF PEEK (PEEK 16398+256*PEE
K 16399)=128 OR 155 THEN PRINT A
T R,E;"█"
 200 LET R=R+(Q>R)-(Q<R)
 210 LET E=E+(W>E)-(W<E)
 220 GOTO 110
 230 LET S=S+1
 240 PRINT AT Q,W;"█"
 250 RETURN
 260 PRINT S
```

CURSO DE ASSEMBLY
aula 3

Z80

Flavio Rossini

O PROGRAMA HEXAMEM: ONDE COLOCAR OS PROGRAMAS EM LINGUAGEM DE MÁQUINA

Um bom lugar para colocar os programas em linguagem de máquina é no fim da memória RAM pois, como já dissemos, o início da memória RAM é reservada para colocar as variáveis do programa interpretador, o programa BASIC e suas variáveis e a tela de TV (Veja pág. 23). Entre as variáveis do programa interpretador está uma chamada **RAMTOP** que indica o fim da memória do computador, ou seja, nos registros da memória correspondentes à essa variável o programa interpretador coloca o endereço que seria do byte imediatamente após o último byte de memória. Esta variável está colocada nos endereços 16388 e 16389, pois sendo ela um endereço, tem 16 bits devendo então ser "quebrada" em duas partes para poder ser armazenada, armazenando-se antes o byte MENOS significativo **(esta regra é usada para todos os dados de 16 bits!)**. Assim digite:

PRINT PEEK 16388 + (256 * PEEK 16389)
(NEW LINE)

(Consulte a aula 1 em caso de dúvida)

Você obtém o valor de **RAMTOP** que dependerá de quanta memória você tem disponível. Se você tiver a expansão de 16K (suposição que será válida para todo o curso) deverá obter:

RAMTOP = 32768 ('8000' em hexadecimal)

o que significa que o endereço do último byte da memória é 32767 ('7FFF' em hexadecimal).

Vamos, então, "enganar" o computador e modificar o valor esta variável com o intuito de reservar o fim da memória RAM para lá colocarmos os programas em linguagem de máquina, sem prejudicar o resto da memória.

Por exemplo, façamos **RAMTOP = 3ØØØØ** ('753Ø' em hexadecimal), lembrando que devemos colocar antes o byte menos significativo ('3Ø'). Digite:

POKE 16388,48 (48 = '3Ø' em hexadecimal)
(NEW LINE)
POKE 16389,117 (117 = '75' em hexadecimal)
(NEW LINE)
NEW
(NEW LINE)

o que corresponde a fazer

RAMTOP = 256 * 117 + 48 = 3ØØØØ

(verifique isto usando novamente a função PEEK para obter o valor de RAMTOP).

Deste modo você reservou a memória 3ØØØØ até 32767 para colocar seus programas no computador, suas variáveis e o conteúdo da tela de TV **NUNCA** irão invadir esta região por "pensar" que a memória termina no

continua

endereço 29999 ('752F' em hexadecimal). Note, todavia, que esta região NÃO será afetada pelo comando NEW e não pode ser passada para a fita através do comando SAVE. Esta última limitação é desagradável e mais adiante veremos como contorná-la.
(OBS.: caso você não tenha expansão de memória use:

```
POKE 16388,173
POKE 16389,67
NEW
```

para reservar a região de 17325 a 18431).

Vamos, a seguir, apresentar um programa em BASIC que coloca na memória RAM um programa em linguagem de máquina elaborada em códigos hexadecimais a partir de um endereço inicial fornecido por você. Lembre-se de **reservar** espaço no fim da memória antes de colocar o programa:

```
POKE 16388,48      (NEW LINE)
POKE 16389,117     (NEW LINE)
NEW                (NEW LINE)
```

A seguir coloque HEXAMEM no computador e vamos introduzir alguns códigos hexadecimais (que por enquanto não significam NADA) dentro do computador! Cada código deve ser seguido de **NEW LINE,** e o programa termina ao introduzir-se a letra P (PARE).

PROG. I.1: HEXAMEM.

```
  1 REM FLAVIO ROSSINI
  5 REM HEXAMEM
 10 FAST
 15 POKE 16388,48
 20 POKE 16389,117
 25 PRINT "ESCOLHA O ENDERECO INICIAL DA"
 30 PRINT "MEMORIA     (>=30000)"
 35 INPUT IN
 40 PRINT
 45 PRINT "MEMORIA INICIAL = ";IN
 50 LET INI=IN
 55 LET A$=""
 60 SCROLL
 65 PRINT "MEM. ";IN;
 70 IF A$="" THEN INPUT A$
 75 IF A$="P" THEN STOP
 80 IF A$="XS" THEN GOTO 0130
 85 IF A$="XF" THEN GOTO 0135
 90 LET AUX=16*CODE A$+CODE A$(2)-476
 95 LET X$=STR$ (16*CODE A$+CODE A$(2)-476)
100 GOSUB 0200
105 PRINT TAB 13;C$;B$;TAB 24;A$( TO 2);TAB (31-LEN X$);X$
110 POKE IN,AUX
115 LET IN=IN+1
120 LET A$=A$(3 TO )
125 GOTO 0060
130 SLOW
135 CLS
140 PRINT AT 3,0;USR INI
145 STOP
200 LET I=1
205 GOSUB 0500
210 LET I=2
215 LET C$=B$
220 GOSUB 0500
225 RETURN
500 IF A$(I)="0" THEN LET B$="0000"
505 IF A$(I)="1" THEN LET B$="0001"
510 IF A$(I)="2" THEN LET B$="0010"
515 IF A$(I)="3" THEN LET B$="0011"
520 IF A$(I)="4" THEN LET B$="0100"
525 IF A$(I)="5" THEN LET B$="0101"
530 IF A$(I)="6" THEN LET B$="0110"
535 IF A$(I)="7" THEN LET B$="0111"
540 IF A$(I)="8" THEN LET B$="1000"
545 IF A$(I)="9" THEN LET B$="1001"
550 IF A$(I)="A" THEN LET B$="1010"
555 IF A$(I)="B" THEN LET B$="1011"
560 IF A$(I)="C" THEN LET B$="1100"
565 IF A$(I)="D" THEN LET B$="1101"
570 IF A$(I)="E" THEN LET B$="1110"
575 IF A$(I)="F" THEN LET B$="1111"
580 RETURN
585 STOP
600 SAVE "HEXAMEM"
610 PRINT "DIGITE BREAK E LIST"
620 PAUSE 300
630 GOTO 600
```

OBS.: Se você NÃO tiver expansão de memória, substitua a linha 3Ø por **PRINT "MEMÓRIA" (> = 17325)",** retire as linhas 95,1ØØ e de 2ØØ a 58Ø, substituindo a linha 1Ø5 por **PRINT TAB 13; A$ (TO 2); TAB 17; AUX.**

Este programa fornece, também, uma "visualização" da memória do computador na tela de TV da seguinte maneira:

MEM.	Nº DA MEM. EM DECIMAL	CONTEÚDO DA MEM. EM BINÁRIO	CONTEÚDO EM HEXADECIMAL	CONTEÚDO EM DECIMAL

Experimente, então colocar alguns nºs hexadecimais na memória a partir do endereço 3ØØØØ. Lembre-se que cada endereço da memória corresponde a 1 byte de dados. Portanto, você deve entrar sempre com um número **par** de dígitos hexadecimais, caso contrário, o programa parará na linha 9Ø. Assim, entre, por exemplo, com o seguinte: (N.L. = **NEW LINE**).

```
RUN (N.L.)
Memória inicial: 3ØØØØ (N.L.)
A4                (N.L.)
B7Ø412            (N.L.)
Ø6                (N.L.)
71FDA1            (N.L.)
ØØCE              (N.L.)
P                 (N.L.)
```

Na tela, você deverá obter:

```
MEM.  30000     10100100    A4   164
MEM.  30001     10110111    B7   183
MEM.  30002     00000100    04     4
MEM.  30003     00010010    12    18
MEM.  30004     00000110    06     6
MEM.  30005     01110001    71   113
MEM.  30006     11111101    FD   253
MEM.  30007     10100001    A1   161
MEM.  30008     00000000    00     0
MEM.  30009     11001110    CE   206
MEM.  30010
```

Com isso, você colocou estes dados da memória 3ØØØØ até a memória 3ØØØ9. Vamos verificar se o programa funcionou. Acrescente, agora, o seguinte programa:

```
3000 REM VERMEM
3005 SLOW
3010 PRINT "FIM = ?"
3015 INPUT FIM
3020 FOR I=30000 TO FIM
3025 SCROLL
3030 PRINT "MEMORIA",I;"        ";PE
EK I
3035 NEXT I
```

NOTA: Se você não tiver expansão de memória, substitua a linha 3Ø2Ø por:

FOR I = 17325 TO FIM

PROG. I.2: VERMEM.

Execute-o fazendo **RUN 3ØØØ** (ou **GOTO 3ØØØ**).

Coloque para a variável FIM o valor 3ØØØ9 (17334 sem expansão), pois colocamos 1Ø dígitos hexadecimais na memória. Ao executar o programa você deverá obter os números previamente colocados, mas em DECIMAL! Portanto, teremos:

MEMÓRIA	**3ØØØØ**	**164**	**(= 'A4')**
MEMÓRIA	**3ØØØ1**	**183**	**(= 'B7')**
MEMÓRIA	**3ØØØ2**	**4**	**(= 'Ø4')**
MEMÓRIA	**3ØØØ3**	**18**	**(= '12')**
MEMÓRIA	**3ØØØ4**	**6**	**(= 'Ø6')**
MEMÓRIA	**3ØØØ5**	**113**	**(= '71')**
MEMÓRIA	**3ØØØ6**	**263**	**(= 'FD')**
MEMÓRIA	**3ØØØ7**	**161**	**(= 'A1')**
MEMÓRIA	**3ØØØ8**	**Ø**	**(= 'ØØ')**
MEMÓRIA	**3ØØØ9**	**206**	**(= 'CE')**

representação em hexadecimal; veja os códigos previamente colocados!

repita este procedimento para outros números hexadecimais até você entender o que está acontecendo. . .

Note que estamos apenas colocando os códigos hexadecimais na memória e mesmo que esses códigos **(A4, B7, Ø4, 12, etc.)** significassem um programa em linguagem de máquina, providencias especiais seriam necessárias para executar o programa.

Estes códigos podem representar um programa em linguagem de máquina, colocado a partir da memória 3ØØØØ. Por exemplo, o código 164 (**A4** em hexadecimal) poderia significar: some os próximos dois bytes (183 e 4); o código 18 (**12** em hexadecimal), compare o resultado com o próximo byte (6) e assim por diante. . .

Note que **NÃO** existem endereços para as linhas de programa, como em BASIC, e as instruções são executadas na sequência em que foram colocadas na memória.

Com todos esses "poréns", quais seriam, então, as vantagens da linguagem de máquina com relação ao BASIC?

A linguagem de máquina é **muito mais rápida** e permite um VISUALIZAÇÃO e um CONTROLE maior do sistema. Em contrapartida, suas instruções são mais simples e limitadas e exigem um cuidado maior na hora da programação.

EXERCÍCIOS

1. Analise o funcionamento do programa HEXAMEM, exceto as linhas 13Ø, 135 e 14Ø.
2. Analise o funcionamento do programa VERMEM.

COMO FAZER SUA ASSINATURA

A nossa revista não será distribuída nas bancas. Para obter seu exemplar mensal, contendo muitos programas para o seu TK, muitas dicas e prêmios interessantíssimos, você deverá fazer uma assinatura: o preço anual da assinatura é de Cr$ 11.800,00. Porém, até 30 de setembro, manteremos o preço de Cr$ 9.900,00, com direito a uma fita inédita de jogos:

São Paulo (1K)
Mansão Maluca (16K)

cujo valor comercial é superior a Cr$ 6.000,00.

Para tanto, você deverá preencher corretamente o cupom anexo, colocá-lo num envelope, junto a um cheque nominal ou vale postal a favor de MICROMEGA PUBLICAÇÕES E MATERIAL DIDÁTICO LTDA., no valor de Cr$ 9.900,00 (oferta válida até 30 de setembro).

O envelope deverá ser selado e endereçado à

MICROMEGA P.M.D. LTDA.
Caixa Postal 60081 — CEP 05096
São Paulo — SP

No verso do cheque escreva:

"Destina-se ao pagamento de uma assinatura (12 números) da revista MICROHOBBY"

Quando este cheque for devolvido ao seu Banco com nosso endôsso, servirá de comprovante provisório até que nosso recibo seja enviado pelo correio.

O QUE ESTÁ RESERVADO PARA O Nº 5:

O SAPO

- **PROGRAMA DO LEITOR:**

  HORA SIDERAL

- **PROGRAMAS DO MÊS:**

  SOMA SINTÁTICA
  TRÂNSITO
  CALEIDOSCÓPIO

- **MUITAS DICAS E INFORMAÇÕES**

- **SEÇÃO NOVA:**

  OS "OITENTAS", PARA O USUÁRIO DO TK ENTENDER O TRS 80 E COMPATÍVEIS NACIONAIS.